Understanding User Cognition: From Spatial Ability to Code Writing and Review

by

Yu Huang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2021

Doctoral Committee:

Professor Westley Weimer, Chair
Professor Stephanie Forrest, Arizona State University
Professor Mark Guzdial
Professor Ioulia Kovelman

Yu Huang
yhhy@umich.edu
ORCID iD: 0000-0003-2730-5077

DEDICATION

The graduate school experience has been a unique journey in my life, which is probably slightly different from most other Ph.D. students in computer science. The most important life wisdom I learned in my graduate school is that you should always be honest about your passion and dreams, and try your best to pursue it. It may not always turn out successful, but at least you will not feel sorry and regretful to your future self about not even giving it a try. During my time in graduate school, I have been through situations including leaving an old program with a Master's degree and starting over in a new major and Ph.D. program, as well as switching research groups and schools. It has been a long journey of endurance, but not a lonely one. There are many people I am sincerely grateful to, without whom I would not have been able to complete my doctoral degree. In this dedication, my words can only express very little of their valuable contribution to lead me to this moment of my life.

First, I would like to thank my advisor, Westley Weimer. Thank you for taking me as your student in spite of my coming from an entirely different field of research. When I joined your group, I did not have a fundamental knowledge of the field, but you generously and patiently spent so much time to help me adapt to the new area and build up my confidence. Thank you for listening to all of my uncertainty about life and future and trusting me all the time. Because of you, imposter syndrome is not that torturous anymore. Thank you for always having faith in me pursuing my career goal (even I did not have faith in myself sometimes) and for giving me all the support and resources you could provide to help me succeed. Without you, I could not even have imagined where and who I am today. Thank you for your professional training on conducting research with impact. I still remember the time when you sat down with me and went through the

code and every single word in our first paper draft. Thank you for helping me improve my English in both professional and life settings. I appreciate that you are always so patient when explaining to me the words, phrases, grammar and sayings that I did not understand. I also appreciate that you always try to actively include me in group discussions on both research and culture topics. As an international student and non-native speaker, these details always warm my heart and make my life in a new country far away from families much easier. Thank you for guiding me how to make life decisions. Without your help, I would not have been able to complete my doctoral degree.

Next, I would like to thank my committee members, Mark Guzdial, Ioulia Kovelman, and Stephanie Forrest. Thank you for being so supportive for this new research approach in software engineering and providing me advice on my research and career. Thank you for your patience on all the graduation procedures that are even more complicated with COVID. I wish you all the best of success in your careers and with your families.

Thank you to Tyler Santander, Xiaosu Hu, Andrew Jahn and to all my collaborators in Psychology and Neuroscience. Thank you for being so patient and tolerating me asking you all the ignorant questions. This is an exciting research direction where we can make a difference together.

Thank you to Denae Ford and Thomas Zimmermann for being my mentors in research, in my career and in my life. Thank you for your generous support of my work and for connecting me to our amazing community. Thank you for letting me have so much fun with you playing Werewolves.

Thank you to Laura Barnes and Benton Calhoun for tolerating me as a student. The happy times I had in your research groups allowed me to get through my early years in graduate school.

Thank you to my undergraduate researchers: Xinyu Liu, Ryan Krueger, Nicholas McKay, Zachary Karas, Michael Flanagan, Ian Bertram and Jack Hong. Thank you for your diligence and hard work. I wish you best of luck.

Thank you to Kevin Angstadt for all your time, help and patience when answering all of my questions. Thank you for tolerating me being ignorant and unobservant. Thank you for helping me learn so much new material and become more confident. Thank you for all the delicious food and

birthday dessert you made for me. I still think the lemon bars you made are the best dessert in the world. I have no doubt that you have an amazingly successful career and life in front of you.

Thank you to Zohreh Sharafi for teaching me how to conduct research using eye tracking. Thank you for comforting and encouraging me when I was in negative moods. I wish you all the best with your career, life and family.

Thank you to Ashley Andreae for handling all the complicated logistics and paper work for my doctoral degree. Thank you for all the administrative support and your eternal patience.

Thank you to Melissa Vaquera for helping me on all the incentive administration for my human studies. Thank you for teaching me how grants work and handling all the reimbursement. My graduate life would have been a mess without you.

Thank you to Dina Salhani for helping me schedule and run all the fMRI experiments. Thank you for chatting with me to fill those dry and empty times between experiments in the fMRI lab. I would not have been able to finish any of the scans without your expertise and patience.

Thank you to my group mates over the years: Johnathan Dorn, Colton Holoday, Madeline Endres, Hammad Ahmad, Fee Christoph, Yirui Liu, Yiannos Demetriou and Annie Li. I truly appreciate our conversations and laughs in the office (and your creativity on exploiting the usage of the stuffed mouse!).

Thank you to all the participants in my studies. I sincerely appreciate your support and participation in the experiments to help advancing the boundaries of human knowledge. Without your support, none of the research effort would have been completed.

Thank you to my friends (Qiushuang, Xiao, Gina, Junwen, Chong, Jingjie, Yuhan, Lingjie, Hao, Xintong, Shibo, Yumeng, Haizhong, Tony, among others) for supporting and encouraging me. Thank you for all the fun we had hiking, watching football, playing board and card games, cooking and doing many other things. Without you, I would not have been able to maintain my mental status, especially during COVID.

I am so fortunate to have loving and supportive families. Thank you to my parents, Zhonglin Huang and Qian Xu, for supporting me throughout my life for all the decisions I made. Thank you

for your selfless love. Thank you to my parents-in-law, Richard and Linda Leach, for giving me so much love and support for my life and career, and for raising up a perfect husband for me. Thank you to my brother-in-law, Eric, for always being so supportive and patient for my husband and me. I want to thank you all for encouraging me to complete this doctorate degree.

Thank you to the love of my life, Kevin Leach. Thank you for your love and support. Thank you for standing by me no matter what happened. Thank you for being so kind-hearted and selfless. Thank you for all the help on research. Thank you for taking care of most of the logistics in our life and absolutely all the yard work. Thank you for tolerating my sometimes-irregular schedule. Thank you for taking such good care of me. You are my best friend and best colleague. I cannot wait for what the future holds for us.

Finally, thank you, the reader, for taking the time to read my thesis. The fact that you are reading this now means that my efforts were worth it.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

ABSTRACT

Understanding how developers carry out different computer science activities with objective measures can help to improve productivity and guide the use and development of supporting tools in software engineering. In this thesis, we present three research components using three different objective measures including neuroimaging (functional magnetic resonance imaging (fMRI) and functional near-infrared spectroscopy (fNIRS)) and eye tracking. We evaluate on over 140 human subjects to explore multiple computing activities, including data structure manipulations, code writing and code review.

First, in a human study involving 76 participants, we examine data structure manipulation and mental rotation tasks using both fMRI and fNIRS. We find a nuanced relationship: data structure and spatial operations use the same focal regions of the brain but to different degrees. They are related but distinct neural tasks. In addition, more difficult computer science problems induce higher cognitive load than do problems of pure spatial reasoning. Finally, while fNIRS is less expensive and more permissive, there are some computing-relevant brain regions that only fMRI can reach. This study paves the way for investigating the foundations of software engineering activities at the cognitive level, as well as providing insights for pedagogical training and guidelines for future studies.

Second, we present a human study in which 30 participants write code and prose while undergoing a fMRI brain scan. Our experiment is the first to use a realistic keyboard that is compatible with modern medical imaging facilities. We find that code writing and prose writing are significantly dissimilar neural tasks. While prose writing entails significant left hemisphere activity associated with language, code writing involves more activations of the right hemisphere, including regions associated with attention control, working memory, planning and spatial cognition. This

study provides a basis for future investigations on complex programming activities. The findings also provide insights to encourage more diversified participation in computer science.

Third, we present the results of a controlled experiment with 37 participants using both fMRI and eye tracking to investigate the neurological correlates of biases and differences between machines (*e.g.*, automated program repair tools) and genders of humans in code review. We find that men and women conduct code reviews differently, in ways that are measurable and supported by behavioral, eye-tracking and medical imaging data. We also find biases in how humans review code as a function of its apparent author, when controlling for code quality. In addition to advancing our fundamental understanding of how cognitive biases relate to the code review process, the results may inform subsequent training and tool design to reduce bias.

This thesis presents a systematic framework and shows that it is possible to conduct studies that acquire objective data in a natural setting to provide an understanding of users' underlying cognitive processes in software engineering tasks. We also provide basic principles and guidelines to adapt multiple psycho-physiological measures to software engineering.

CHAPTER 1

Introduction

With software-related innovations driving a $3.8 trillion global IT market [1; 2] and demand for university computer science courses outstripping the supply of professors [3], the value of software engineering is increasing rapidly. Modern software engineering is not just about programming, but is also about understanding how and what to program, how to structure information, how to plan the work, how to lead the people and how to get them to communicate and collaborate effectively [4]. In recent years, modern software systems are also being developed by increasingly distributed and diverse teams. For instance, Open-Source Software (OSS) has established itself not only as a critical resource in today's software infrastructure and economy, but also as a launching point for careers in technology [5]. From the early stages of talking to customers and clients (called requirements elicitation) and planning the entire project design (called system specification), all the way through to the maintenance of software systems after deployment, a fundamental understanding of multiple activities is essential to improving productivity and efficiency in modern software engineering. This importance is already emphasized in the software industry, with Fortune 500 companies, such as Amazon and AT&T, committing massive resources to retrain up to half of their workforces to obtain better productivity and efficiency in programming-intensive areas [6; 7] as of 2020.

Over the years, among research approaches in modern software engineering, some solutions worked well while others did not (*e.g.*, aspect-oriented programming vs. object-oriented programming [8], system design decisions and usability across different demographic groups [9], etc.).

However, we lack a grounded theory and supportive information to understand and explain these failures and successes. Such a theory could make it possible to design software engineering techniques in a more efficient way. Despite the increasing prevalence of software and demand for skilled programmers, in the past, researchers primarily relied on methods that collect subjective information and opinions from users (*e.g.*, think-aloud protocols, questionnaires, surveys, and interviews, etc.) to study software engineering tasks [10; 11; 12]. Although these traditional methods contribute important evidence and advance the state of the art, they suffer from the Hawthorn (observer) effect [13; 14] and may not be reliable [15; 16; 17; 18]. For example, researchers may collect users' answers to study their behaviors or design preferences by asking questions such as "which product do you think is better", where human bias related to the racial background of the interviewers (among other qualities) can harm the reliability of the results [19]. To complement and enhance data collected using such traditional methods, and mitigate biases introduced therein, we favor objective measures to provide insights into the cognitive processes that underlie various software engineering activities.

In Psychology, *cognition* is the ability to process information through perception, or the accumulation of information that we have acquired through learning or experience. It includes different *cognitive processes*, like learning, attention, memory, language, reasoning, decision making, etc., which are part of our intellectual development and experience (*e.g.*, [20]). Cognitive processes have attracted significant attention in many academic disciplines (*e.g.*, neurology, psychology, anthropology, philosophy). In cognitive psychology, researchers have studied cognitive processes since the 1950s and the interest in research about cognition has increased since the 1960s. One example is *spatial ability*, the manipulation of three-dimensional shapes in imagination, which psychologists have shown to be a major factor in proficiencies such as mathematics [21; 22], natural sciences [23; 24], engineering [25], meteorology [26], and map navigation [27]. At the same time, the emergence of technologies that look inside neurological processes, like medical imaging and sensing technologies, such as *functional magnetic resonance imaging (fMRI), functional near-infrared spectroscopy (fNIRS)*, and eye tracking, has advanced and contributed to the neurological

and cognitive understanding of thinking processes.

Researchers in Psychology have studied cognitive processes of different tasks, ranging from musical performance [28], to food cravings [29] to prose writing [30; 31; 32]. However, these studies rarely involve computer science tasks. There is also a significant body of work investigating the psychology of programming, ranging from the cognitive prerequisites of programming [33], to programming behaviors [10; 34] to entire theories of the coding process [12]. Unfortunately, these studies usually rely on self-reporting data which may lack foundational evidence. To address the concerns associated with self-reporting, researchers in software engineering have turned to medical imaging to obtain neurological evidence regarding programming activities. Research using medical imaging techniques has examined brain patterns in code comprehension, code readability and bug detection [35; 36; 37; 38]. However, these studies fail to provide an understanding of cognitive processes for higher-level and more industry-related activities, such as reviewing others' code and writing programs, as well as the differences between diversified groups of users. Some studies in software engineering explored individual differences in outcomes (*e.g.*, associated with gender) [39; 40; 41], but they do not illuminate how user demographics actually affect decision making in programming tasks.

In this thesis, we present a systematic framework to objectively measure and understand user[1] cognition in software engineering activities by introducing three research components that range from foundational to high-level tasks in modern software engineering: data structure manipulation, code writing, and code review. We also introduce the design and methodology for studying cognitive processes in these activities with multiple state-of-the-art objective measures, including fMRI, fNIRS and eye tracking. This thesis shows that it is feasible to study user cognition in software engineering activities and reveal truths that may be overlooked by traditional methods such as self-reporting.

[1]For a broader group of audience, "user" usually refers to end users. In this thesis, "user" refers to humans that are involved in any type of software engineering activities. Broadly, the framework presented in this thesis can be applied to both end users and programmers.

1.1 Approach

While findings relating to cognitive processes have successfully transitioned to guiding behavioral and developmental improvement in domains like mathematics [42] and education [43], we still lack a fundamental understanding of the cognition behind software activities. Such an understanding would help programmers, academics, and industry participants: from understanding productivity and expertise [44; 45; 46] to increasing participation in the modern workforce [47; 48] to guiding pedagogy [49; 50] to augmenting unreliable self-reporting [51; 52]. To obtain such an understanding, we present a systematic framework that satisfies the following criteria:

- **Objective Measures.** It is critical to measure the relevant factors objectively in computer science tasks. Research in both Psychology and Computer Science has shown that subjective measures may not be adequately trustworthy (*e.g.*, [15; 16; 17; 18]). Objective measures are necessary to understand user cognition in a generalizable and reliable way.

- **Foundational Understanding.** A fundamental understanding of the cognitive processes in programming tasks is essential to help users solve software problems, learn programming, and make decisions in software development, as well as to further improve productivity and efficiency [53; 54].

- **Higher-Level Tasks.** We desire an understanding of higher-level programming tasks. Software development is a complicated process consisting of different components. In the software industry, development usually includes higher-level tasks such as code reviews [55; 56]. The efficiency of such higher-level tasks directly and significantly affects their time and monetary cost [57; 58]. Thus, it is important to understand the cognitive processes for these semantically-rich and industry-related activities.

- **Generalizability Across Users.** We desire an understanding that applies to a wide range of people. Modern software development is conducted in an environment of diverse populations and diversity is important to effectiveness in software engineering [59; 60]. The diversity of

the CS workforce has been receiving more attention over time [61; 62] as of 2020. Thus, we desire a theory that accounts for demographic differences (*e.g.*, gender) across users.

The overall thesis statement of this dissertation is:

It is possible to meaningfully and objectively measure user cognition to understand the role of spatial ability, fundamental processes and stereotypical associations in certain software engineering activities by combining medical imaging and eye tracking.

We combine several insights to form the basis of a systematic study for understanding user cognition in computer science. **First**, with the increasing emergence of *medical imaging* and *eye tracking* (*e.g.*, fMRI, fNIRS, and eye trackers), it is now possible to conduct studies that acquire objective data in a natural setting to provide an understanding of the underlying cognitive processes of certain tasks. For instance, modern medical imaging techniques allow researchers to investigate the neurological patterns in human brains during different tasks. **Second**, we can adapt *scientific approaches and concepts from other domains* to assist our investigation and understanding for computer science activities. For example, education researchers have studied the influence of spatial ability and shown that it is a major factor in proficiencies such as mathematics [21; 22] and the natural sciences [23; 24]. They have also designed corresponding interventions to enhance spatial ability [63; 64; 65; 66]. These results inspire us to study the relationship between spatial ability and programming to help with programmers' productivity. **Third**, it is now possible to study historically-subjective factors like cognition by designing *rigorous controlled experiments*. For instance, contrast-based experiments are widely used in medical imaging studies to investigate the brain. Such designs make it possible to focus on brain activities that are only relevant to the actual experimental conditions. These three insights support our systematic framework that satisfies all four of the desired properties we require.

To understand cognition in software engineering activities, this thesis presents the results of three studies that investigate certain relevant and important behaviors of programmers.

Our first research component is to investigate the relationship between data structure manipulation and spatial ability using multiple medical imaging techniques. Studies in Psychology have displayed the importance of, and possible interventions for, enhancement for spatial ability in several domains [22; 26; 64; 65], but it is rarely studied within software development. We hypothesize that spatial ability is highly associated with software tasks on a foundational level. If so, we can leverage training experience for spatial ability in Psychology to improve pedagogy and productivity in computer science. We use medical imaging to compare the brain activities of spatial ability and data structure tasks because (1) data structures are a fundamental element in computer science that affect the performance and cost of many systems, and (2) medical imaging techniques can provide us with a neurological basis for the relationship between two tasks. We statistically validate this relationship on humans using two medical imaging modalities.

Our second research component investigates the relationship between code writing and prose writing using medical imaging. The goal of this study is to understand the cognitive process of code writing, a crucial activity in software engineering. We use prose writing as a baseline to ground our results. While some studies have explored how software developers read code (*e.g.*, [35; 67]), there is no research studying the cognitive processes of creativity in programming such as code writing. One challenge is that normal keyboards may not be safely deployed with state-of-the-art medical imaging. We design a bespoke magnet-safe keyboard to allow typing and editing. We test the brain activity relationship between code writing and prose writing using statistical tests on human participants.

Our third research component is to study the decision making process in code reviews using medical imaging and eye tracking. The goal of this study is to investigate the effects of stereotypical associations (*i.e.*, bias) in software engineering. There is research showing that software developers do not recognize this potential bias when checking the source of code in code reviews [41] and female developers have lower acceptance rates when their identities are directly recognizable in open source projects [40]. We use open source projects and controlled author information to study the cognitive processes in code reviews. In addition, we also monitor subjects'

eye movement using eye tracking devices. We compare the brain activities and eye motions across conditions using statistical tests.

1.2 Summary and Organization

The main contributions of this thesis are the following:

- A mathematical model and analysis of the particular relationship between data structure manipulation and spatial ability based on fundamental medical imaging results.
- A mathematical model and analysis of the particular cognitive processes involved in the higher-level software engineering activities of code writing and code review (including the effects of certain human biases).
- A comparison of cost-benefit and feasibility tradeoffs between different medical imaging techniques for software engineering.

The remainder of this thesis is organized in the following manner. In Chapter 2, we provide relevant background material on the formalisms and techniques used in the remainder of this dissertation, including the general introduction of medical imaging, mechanisms of fMRI, fNIRS, eye tracking and contrast-based analysis, and processes of software engineering tasks. We also provide related work for studies presented in this thesis in the end of Chapter 2. In Chapter 3, we present the study of understanding user cognition in data structure manipulation using two medical imaging modalities (*i.e.*, fMRI and fNIRS) and the comparison between the two modalities for future research principles. In Chapter 4, we present the study of investigating the cognitive processes in a more complex and higher level software engineering activity, code writing, and introduce a bespoke keyboard that allows typing in modern medical imaging facilities. In Chapter 5, we present the study on exploring biases in code review across different user groups with both medical imaging (*i.e.*, fMRI) and eye tracking. In Chapter 6, we summarize the work in this thesis and present

discussion on potential future research directions in the community based on results presented in this thesis.

CHAPTER 2

Background and Related Work

Prior to commencing our exploration of understanding user cognition in computational activities, we first introduce key concepts, results and techniques related to psycho-physiological measures for a computer science audience in Section 2.1. Second, we review the psychology and software engineering tasks involved in this thesis, including the study of mental rotation in psychology, supporting our experimental use of it as a neurological basis for spatial ability, code comprehension, code and prose writing, and code review. Finally, we introduce related work that has been done in software engineering with various psycho-physiological measures.

2.1 Psycho-physiological Measures

Human brains run many types of operations to center all information collected to effectively operate in the world. In software engineering, similarly, developers' cognitive processes involve the acquisition, storage, interpretation, manipulation, transformation, and use of relevant knowledge. When developers think and reason about a task, usually their visual attention (*i.e.*, the information they focus on visually) indicates the information they are acquiring and certain regions of their brains are activated. Developers' visual attention can be captured through their eye movements via techniques such as eye tracking (described in detail in Section 2.1.3). When an area of the brain is in use, blood flow to that region increases accordingly. The theory behind the process is, neurons do not have internal reserves of energy in the form of sugar or oxygen, so their firing causes a

need for more energy to be brought in quickly [68]. Developers' brain activation patterns can be measured through neuroimaging techniques (described in detail in Section 2.1.1).

In this section, we overview the mechanism and research use of the two popular neuroimaging techniques: functional magnetic resonance imaging (fMRI) and functional near-infrared spectroscopy (fNIRS), as well as eye tracking, including their relative advantages and disadvantages for our experiments to measure user cognition. We also introduce the experimental design principles for neuroimaging techniques.

2.1.1 Neuroimaging

As mentioned above, human brains support neural activities with energy provided in blood oxygen. *Functional neuroimaging* techniques are used to study brain activity based on this theory. In this thesis, we also refer to neuroimaging as medical imaging for convenience and the significant overlap between them [69]. Over the past 30 years, non-invasive *in vivo* functional neuroimaging techniques have emerged as important tools in understanding cognitive processes. The most popular of these techniques, fMRI, and its counterpart, fNIRS, provide several advantages.

First, as non-invasive tools, fMRI and fNIRS pose significantly less risk and can access a wider range of brain regions than previous invasive techniques (*e.g.*, electrocorticography (ECG)). Second, fMRI and fNIRS provide a wider field of view and higher spatial resolution than other functional neuroimaging techniques (*e.g.*, electroencephalogram (EEG), Magnetoencephalography (MEG)), allowing for the characterization of a brain region's contribution to a specific task. Third, fMRI and fNIRS avoid the use of ionizing radiation or radioactive elements that is common in many other neuroimaging modalities (*e.g.*, computerized tomography (CT), positron emission tomography(PET)). Instead, fMRI and fNIRS rely on the *hemodynamic response*, the metabolic changes (*e.g.*, oxygen, glucose) in neuronal blood flow to active brain regions, using oxygen consumption as an indirect measurement for brain region activity [70] (see Sections 2.1.1.1 and 2.1.1.2 for more detail).

In neuroimaging, researchers adapt anatomical classification systems that divide the human

brain into areas, each associated with specific neurological functions. A popular choice is the Brodmann anatomical classification system which divides the brain into 52 areas [71]. In this thesis, we use the Brodmann system and refer to brain areas with the notations of BA1–BA52.

In the remaining of this subsection, we will first introduce the mechanisms of fMRI and fNIRS. Then we will compare fMRI and fNIRS from the perspectives of research applications and setups.

2.1.1.1 How fMRI Works

Through a process called the *hemodynamic response*, blood releases oxygen to active neurons at a greater rate than to inactive neurons. This causes a change of the relative levels of *oxyhemoglobin* and *deoxyhemoglobin* (oxygenated or deoxygenated blood) that can be detected on the basis of their differential magnetic susceptibility [68]. In neuroscience, we refer to the contrast between oxyhemoglobin and deoxyhemoglobin as the *blood-oxygen-level-dependent* (BOLD) signal [72]. Hemoglobin has different magnetic properties in its oxygenated and deoxygenated forms (deoxygenated hemoglobin is *paramagnetic* and oxygenated hemoglobin is *diamagnetic*), which leads to magnetic signal variation [68]. Thus, certain neuroimaging techniques such as fMRI can be used to detect brain activities. fMRI measures BOLD signals via the application and removal of a series of magnetic fields. The energy that nuclei emit upon returning to their original positions can be used to determine their locations. As task-related brain activity is mapped onto an anatomical scan of the participant's brain in the associated mathematical analysis, participants must lie still in the narrow fMRI machine throughout the experiment with minimal head movement. For a more detailed introduction and explanation of the foundational physical and physiological principles of fMRI, the reader is referred to Hashemi *et al.* [73] and Ulmer *et al.* [74]. For a high-level example of an fMRI machine, please refer to Chapter 3.2.3.

2.1.1.2 How fNIRS Works

Similarly, oxygenated and deoxygenated hemoglobin also lead to optical differences. Thus, techniques, such as fNIRS, can also be used to detect brain activities, where light signals of near-

infrared wavelengths are reflected differently according to the ratio of (de-)oxygenated blood. fNIRS also measures the hemodynamic response to determine active brain regions. fNIRS relies on differences in the absorption of chromophores, groups of atoms that generate color through the absorption of light, between oxygenated and deoxygenated hemoglobin. Light is emitted and detected through devices placed at specific locations on a scalp cap worn by the participant. Unlike fMRI, fNIRS measures concentration changes in oxygenated and deoxygenated hemoglobin separately. fNIRS admits relative freedom of motion and has few environmental restrictions. For example, participants can sit in front of a standard computer and monitor and perform in a more realistic software development setting. For more detailed introduction and explanation of the foundational physical and physiological principles of fNIRS, please refer to Ozaki *et al.* [75]. For a a high-level example of fNIRS equipment and the associated setup, please refer to Chapter 3.2.3.

2.1.1.3 Comparison of fMRI and fNIRS

Both fMRI and fNIRS have been widely used in psychological and clinical research to develop a deeper understanding of brain functions such as sensory, verbal, and motor processing [76; 77; 78; 53; 79; 80]. As a result, fMRI and fNIRS are popular in research. In 2020 alone, there are more than 37,000 publications on fMRI officially collected in PubMed only. Among other examples, fMRI has been used to study face recognition, decision making, resting, and vegetative states [81; 82; 83; 84; 85; 86]. Similarly, the use of fNIRS is also on the rise [87]. The applications of fNIRS span many fields, such as behavioral development, psychiatric conditions, and brain injury [87; 88; 89; 90].

However, fMRI and fNIRS rely on the hemodynamic response (see Section 2.1.1.1), and share several limitations. One limitation arises from *hemodynamic lag*: the onset of changes in neuronal blood flow peaks several seconds after the onset of stimuli [91; 92]. Similarly, the hemodynamic response saturates over time [54], resulting in weaker signals for tasks involving sustained activity. The hemodynamic response enforces experimental restrictions such as lower and upper limits on stimuli (commonly 30 seconds, no longer than 60 seconds), as well as demanding robust mathe-

matical analysis [93; 94].

fMRI provides excellent spatial resolution and deep penetrating power. It is a precise neuroimaging modality that captures activations across the whole brain. In contrast, fNIRS provides inferior spatial resolution and depth compared to fMRI due to inconsistent photon paths and the limited penetration of near-infrared light. As a result, fNIRS also provides a noisier signal, leading to more careful considerations in experiment and analysis design. Likewise, fNIRS places a burden on the researcher to decide, in advance, on the placement of light emitter-detector devices. Given finite placement space on the scalp, the number of regions fNIRS can measure simultaneously is limited. However, as of 2021, fNIRS is gaining traction as a neuroimaging technique due to its portability, ease of administration, ecological validity, and lower cost. In contrast, the high cost, restrictive environment, and high sensitivity to participant motion of fMRI limit its practicality. In this thesis, we present recommendations for the use of fMRI and fNIRS to study software engineering in Chapter 3.

2.1.2 Basic Principles for Neuroimaging Experimental Design

In this section, we will introduce basic principles in study design using neuroimaging techniques. The design principles are discussed from the perspective of fMRI, while experimental design for fNIRS shares the same considerations. More discussion can be found in Amaro Jr. *et al.* [95].

2.1.2.1 Stimulus Representation

Initially, fMRI studies relied on sequentially presented stimuli within blocked conditions with a long history relevant to an historical influence of PET studies: researchers had investigated changes in blood flow measured over time periods of up to one minute when the human participants had to maintain their cognitive engagement. Over the last decade, fMRI has employed a variety of presentation schemes [95].

Block Design. In a *block design*, two or more conditions are alternated sequentially [96]. Each block will have only one experimental condition presented. After each block, there is a rest condition in which the hemodynamic response has enough time to return to baseline. Thus, a maximum amount of variability is introduced in the signal and this allows block design to offer considerable statistical power [97]. However, block design is limited by constraints such as signal drift and head motion (see Chapter 3.3.1 for techniques to mitigate such limitations).

Event-related Design. In contrast to block designs, the presentation of *event-related designs* is randomized and the time in between stimuli can vary [96]. Event-related designs model the change in fMRI signal in response to neural events associated with behavioral trials. Within each trial, there are a number of events such as the presentation of a stimulus, delay period, and response. Event-related designs allow more real world testing, however, the statistical power of event related designs is inherently low, because the change in the BOLD signal following a single stimulus presentation is small [98]. The disadvantages of event-related design also include more complex design and analysis.

Due to the difference in statistical power, in this thesis, we employ block design for the presented studies.

2.1.2.2 Contrast-based Design and Analysis

Both block and event-related designs are based on the *contrast-based design* (also called the subtraction paradigm), which assumes that specific cognitive processes can be added selectively in different conditions [95; 96]. Any difference in the BOLD signal between two conditions is then assumed to reflect the differing cognitive process. By making the conditions differ in only the cognitive process of interest, the fMRI signal that differentiates the conditions should represent this cognitive process of interest [98].

Following the convention in Psychology and Neuroscience, in this thesis, we use the notation $\mathsf{A} > \mathsf{B}$ to refer to the *contrast* (or difference) between two task conditions. For example, $\mathsf{A} > \mathsf{B}$

refers to the comparison of brain activations during task A vs. task B. Contrasts are *directional* tests: the aforementioned $\mathsf{A} > \mathsf{B}$ contrast will specifically attempt to identify regions in which average task A activity is *greater* than B. Critically, this does *not* imply that the inverse contrast ($\mathsf{B} > \mathsf{A}$) will reveal regions in which task B activity is significantly greater than task A activity, as differences in the opposite direction may be too small to be statistically meaningful (particularly with the conservative thresholds we use to guard against false positives).

2.1.3 Eye Tracking

When users conduct software engineering activities, for example, reading code, their eye movements indicate the real-time visual attention. Such eye movements can be measured with eye trackers. Modern eye trackers are non-invasive, versatile, easy-to-use devices that have been used to study diverse topics, such as surgery [99], driver-vehicle interfaces [100], human-computer interactions [101], gaming [102; 103], and software engineering [104; 105].

Eye trackers are designed to collect a participant's visual attention by recording eye-movement data [106]. Visual attention triggers the mental processes required for comprehending and solving a given task, while cognitive processes guide the visual attention to specific locations. Thus, eye tracking provides useful information to study the participant's cognitive processes and effort while performing tasks [107]. Compared to conventional self-reporting methods, eye trackers are a cost-effective way of collecting data at a fine level of details with minimal intrusion [108]. An eye tracker also provides information that is not available from conventional methods, including fine-grained patterns of visual attention (visual attention trends) [109; 110]. A *visual attention trend* encapsulates changes in participant's visual attention over time.

Raw data recorded by an eye tracker is processed by an event detection algorithm and results in *eye gaze* data. Eye gaze data is studied with respect to certain *areas of interest* (AOIs) in a stimulus. AOIs are manually defined by the experimenter based on research questions and variables [101; 109; 111; 112].

Eye gaze data is typically divided into two categories [106] based on ocular behavior. A *fixa-*

tion is a spatially-stable eye gaze that lasts for approximately 200–300 ms (on average, three eye fixations happen per second during active looking). During a fixation, visual attention is focused on a specific area of display. Researchers in psychology claim that most of the information acquisition and processing occur during fixations [101; 109] and that a small set of fixations suffices for the human brain to acquire and process a complex visual input [106; 109; 111]. Fixation data has been extensively used to measure the visual effort (cognitive load) representing the tasks and stimuli being assessed [101; 112; 113]. Longer fixation duration and higher number of fixations indicate higher visual effort [101; 113]. A *saccade* is a continuous and rapid eye-gaze movement that occurs between fixations. Saccadic eye movements are extremely rapid (within 40–50 ms). Cognitive processing during saccades is very limited [106; 109]. In this thesis, we focus on fixation data to measure users' visual attention.

How Eye Tracking Works. Modern, non-intrusive eye trackers consist of two miniature cameras and one infra-red light source. They measure and track the human eye's focus point using the "corneal-reflection/pupil-center" method [107; 114]. The invisible infra-red light is directed into the participant's eyes. After entering the retina, a large proportion of the emitted light is reflected back and creates a strong reflection which causes the pupils to appear very bright. A corneal reflection is also generated and shown as a sharp glint over the iris. Cameras then record the center of the pupil and location of the corneal reflection while image processing identifies and tracks the eyes.

Neuroimaging techniques provide information on the brain activation patterns which indicate the thinking process on the neurological level. At the same time, eye tracking allow researchers to measure visual attentions involved in certain tasks. Though there is a long way to go to completely decode user cognition in software engineering tasks, these modern psycho-physiological measures allow us to further understand it with objective measurement. In this thesis, we will use fMRI, fNIRS and eye tracking, to measure and understand the cognitive process involved in various software engineering activities. In the following sections, we will introduce several critical software engineering and Psychology tasks to investigate in this thesis with the psycho-physiological mea-

sures introduced in this section.

2.2 Psychology Tasks

In this section, we introduce tasks that have been studied in Psychology and are also relevant to software engineering activities we investigate in this thesis. These psychology tasks include *mental rotation* and *prose writing*.

2.2.1 Mental Rotation

Mental rotation is defined as the capacity to quickly and accurately rotate two- or three-dimensional figures in imagination [115]. Mental rotation tasks generally involve comparing two three-dimensional objects rotated about an axis, and are a standard paradigm for testing spatial ability [116]. There are many interpretations of spatial ability, including the determination of spatial relationships between objects and the mental manipulation of spatially-presented information. Despite spatial ability's influence in a wide range of disciplines, it has rarely been studied within software engineering. To the best of our knowledge, only one previous study (conducted by Aharoni [10]) focused on the relationship between software engineering tasks and spatial ability. This previous work relied on interviews with students to understand their thought processes and suggested that programmers use visual representations to reduce the level of abstraction of data structures. However, no quantitative relationship has been investigated.

Neuroimaging has provided evidence that mental rotation involves the right parietal lobe, a region believed to be responsible for spatial ability [117; 118; 119]. In our experiments we use mental rotation as a validated test case for spatial ability. Shepard and Metzler found that the time required to solve mental rotation tasks is a linearly-increasing function of the angular difference between the orientations of the two objects [120]. Gogos *et al.* studied the difficulty of mental rotation using fMRI to identify rises in the BOLD signal with increased angles of rotation [121]. Mental rotation is a meaningful comparison for the investigation of difficulty in our

studies. Psychology research has shown spatial ability to be a major factor in proficiencies such as mathematics [21; 22], natural sciences [23; 24], engineering [25], meteorology [26], and map navigation [27].

In this thesis, we will relate data structure manipulation (*i.e.*, operations on data structures) and spatial ability (measured via mental rotation tasks) in Chapter 3, to investigate fundamental activities in software engineering.

2.2.2 Prose Writing

For the purpose of this thesis, *prose writing* refers to any natural language writing work that follows a basic grammatical structure, such as arranging words and phrases into sentences and paragraphs [122]. Theoretically, prose writing can refer to many types of natural languages. In this thesis, we focus on English prose writing. There is a significant body of research in Psychology to study the cognitive processes of prose writing (see Berninger and Winn [30] for a survey).

Early research dedicated to the cognitive processes of prose writing was conducted without medical imaging. Hayes and Flowers proposed a theory of the cognitive processes of writing in 1981 [123]. Research in the field continued throughout the 1980s and 1990s, focusing on more nuanced aspects of prose writing cognition including second-language proficiency [124; 125] and studies on gaining writing expertise (*e.g.*, [126]). Beaufort later proposed a social apprenticeship model for gaining writing expertise, highlighting a continuum of novice to expert writing roles as an opportunity for such a framework [126].

Unlike code writing, researchers have since leveraged medical imaging to establish objective models of the prose writing process and have used such an understanding to improve prose writing as a whole. Menon and Desmond were among the first to use fMRI to understand prose writing. Their study, in which participants wrote by dictation in an fMRI machine, found activation in only the left hemisphere, particularly the left superior parietal lobe [31]. Our study similarly found activation in left temporal region for prose, but also found the right temporal region to be associated with code writing. Shah *et al.* later used fMRI to study the neural correlates of creative writing

with an experiment that separated the "brainstorming" phase of prose writing from "the act of writing a new and creative continuation of a given literary text" [127].

There has also been particular interest in studying the specialization of writing-specific brain regions. Sugihara *et al.* studied the brain's writing center during both left- and right-handed writing tasks [128], identifying regions crucial to the core process of writing. Planton *et al.* later identified brain regions that are consistently involved in prose writing tasks, as well as differences in brain activation across writing, drawing, and oral spelling [129]. Similarly, Purcell *et al.* studied the neural basis of spelling and its relation that of reading: their study used a QWERTY keyboard to study *prose* writing with fMRI [130]. However, their experimental design was restricted to typing single words by dictation and without the participants having any live feedback while typing.

Historically, the development of a fundamental, neurological understanding of other activities, such as prose writing, has proved useful as a guide in pedagogy and research. Berninger and Winn credit advanced brain-imaging technologies as the primary development near the end of the 20th century that reformed prose writing research and education [30]. Examples of brain imaging contributing to pedagogy include the use of verbal and non-verbal cues and strategies to improve learning [131; 132], as well as teaching such cues and strategies to overcome inefficiencies in temporally-constrained verbal working memory [133].

In comparison, researchers lack a corresponding fundamental understanding of code writing that might illuminate new ways to improve code writing skills. Our thesis is motivated by the belief that such a foundational understanding could guide more focused training and teaching strategies for code writing.

2.3 Software Engineering Tasks

In this section, we present some relevant background on the software engineering tasks considered in this thesis, as well as results related to expertise and imaging.

2.3.1 Data Structure Manipulation

In computer science, a *data structure* is a particular way of organizing data in a computer so that it can be used effectively. Manipulating data structures is one of the most fundamental skills required in software development. For example, programmers use data structures such as *lists*, *arrays* and *trees* to store data and apply operations such as *sort*, *insert* and *traversal* to the data structures. As of 2021, only one other previous line of research has considered data structures at a cognitive level. In a qualitative study involving nine computer science majors, Aharoni investigated student thought processes when dealing with data structures [10; 134]. Aharoni found that visual representations influenced students' perceptions of the overall properties of data structures, suggesting that programmers use visual representations to reduce levels of abstraction. While we draw inspiration from Aharoni's investigation of data structure mental manipulations, rather than focusing on qualitative self-reporting, our studies use objective measurements of associated visual and neural representations.

2.3.2 Code Comprehension

Much research, both recent and established, has argued that reading and comprehending code play a large role in software maintenance [135]. A well-known example is Knuth, who viewed this as essential to his notion of Literate Programming [136]. He argued that a readable program is "more robust, more portable, [and] more easily maintained."

Knight and Myers argued that a source-level check for readability improves portability, maintainability and reusability and should thus be a first-class phase of software inspection [137]. Basili *et al.* showed that inspections guided by reading techniques are better at revealing defects [138]. An entire development phase aimed at improving readability was proposed by Elshoff and Marcotty, who observed that many commercial programs were unnecessarily difficult to read [139]. A 2012 survey of over 50 managers at Microsoft found that 90% of responders desire "understandability of code" as a software analytic feature, placing it among the top three in their survey [140, Fig. 4].

2.3.3 Code Writing

There is strong interest from both academia and industry in improving programmers' ability to write code [141; 142; 143; 144]. Developing methods to make code writing more productive, accurate, and accessible has long been a software engineering goal. Such methods are often motivated by an understanding of programming psychology, such as block-style languages designed for younger ages (e.g., Scratch [145]) and IDEs intended to increase productivity [146].

Researchers from as early as the 1950s have sought to understand the psychology of writing code. Early efforts focused on cognitive load [147], psycholinguistic theory [148], and expertise [149; 150], among other topics. In 1977, Brooks proposed an entire theory of programming behavior oriented toward explaining transcriptions of participants asked to talk aloud while performing programming tasks [12]. More recent work includes studies on how experts and novices classify algorithms [151; 152] and programmers' use of the Web when writing code [153].

In contrast to this thesis, previous studies have not directly investigated the cognitive processes of code writing on the neurological level.

2.3.4 Code Review

Static program analysis methods aim to find defects (or other critical information) in software and often focus on discovering those defects early in the code's lifecyle. At its core, *code review* is the process of developers reviewing and evaluating source code content and changes. Code review is one of the most common forms of static analysis as of 2021 [154]; well-known companies such as Microsoft, Facebook, and Google employ code review regularly [155; 156]. Typically, the reviewers are someone other than author of the code under inspection. Code review is often employed before newly-written code can be committed to a larger code base. Reviewers may check for style and maintainability deficiencies as well as defects. Numerous studies have affirmed that code review is one of the most effective quality assurance techniques in software development [157; 158; 159; 160]. While it is a relatively expensive practice due to high developer input, it successfully identifies defects early in the development process. This benefit is valuable because

the cost to fix a defect generally increases with the time it goes unnoticed [161; 162; 163].

Because code review is a critical software development activity, which directly and largely affects the quality and maintenance cost of software design, it is essential to understand the limitations of current code review process and improve the efficiency and effectiveness of code review.

2.3.5 Gender Biases and Differences in Software Engineering

Previous studies have found that the field of software engineering has very low participation from women [164]. This is in spite of multiple studies that have found a positive correlation between team diversity and team performance in this field [59; 60; 165]. Several candidate explanations for low participation among women have been proposed in multiple studies: for example, women in software engineering (and, more generally, in male-dominated fields) tend to see more criticism on the quality of their work, more rejection of work, more harassment in the workplace, lower chances of promotion, and more ridicule for both success and failure than men [166; 167; 168; 169; 170; 171; 172]. While there has been extensive research into the measurement of and the social causes for these biases, there has been no research into the psychological basis behind code review decisions as of 2021. Because of the importance of code review, we seek to avoid potential bias on behalf of the reviewer to make code review as effective as possible.

Knowing that gender-based bias and discrimination exist in software engineering, in this thesis, we hope to understand and mitigate its impact on code review.

2.4 Medical Imaging and Eye Tracking in Software Engineering

In this section, we discuss previous work related to computer science and neuroscience, as well as studies in a wider range of domains that have used fMRI and fNIRS. Additionally, we discuss previous research on eye-tracking studies in software engineering.

We note that the use of medical imaging in software engineering is still exploratory; between

2014 and 2021, there have been fewer than 20 publications that have studied its associated cognitive processes with either fMRI or fNIRS [35; 67; 173; 174; 38; 37; 36; 175; 176]. Given the tradeoffs between these two neuroimaging techniques, the community has not settled on the best option for studying software engineering tasks. Siegmund *et al.* introduced the study of software engineering tasks with fMRI, focusing on code comprehension [177]. Their analyses identified five brain regions with distinct activation patterns, all of which are relevant to working memory, attention, and language processing. Newer work has explored the relationship between comprehension, code and prose review with expertise [67], bug detection and brain activities [36; 37], code comprehension with eye tracking [176] and the effects of beacons (semantic cues) on code comprehension [175]. In this thesis, we apply Siegmund *et al.*'s innovative use of neuroimaging, and adopt these previously-identified brain regions as an established basis for verbal processing in software engineering.

Similar to fMRI, fNIRS has been used to study the relationship between program comprehension and brain activity. Researchers used NIRS signals and found an increase in cerebral blood flow when analyzing obfuscated code and code that requires variable memorization [173; 174]. Subsequent research studied the effect of code readability on cognitive load [38; 175].

Besides fMRI and fNIRS, researchers have tried other medical imaging tools to study software engineering. Crk *et al.* used electroencephalography (EEG) to investigate the role of expertise in programming language comprehension. Their study found that the brain's electrical activity can indicate both prior programming and self-reported experience levels [178]. Lee *et al.* used EEG in a similar setting [179] to Floyd *et al.*'s work [67]. Parnin used electromyography (EMG) to explore the roles of subvocalization for different programming [180]. Researchers have explored the link between programming tasks and cognitive load [181; 182] using EEG, EMG, and eye tracking.

While medical imaging is relatively new to the software engineering community, it has made remarkable contributions in guiding behavioral enhancement and development in different domains, such as mathematics [42] and education [43]. For instance, cognitive understanding of numeracy has inspired researchers to use different measures to predict individual differences in mathemat-

ical development and achievement [183; 184; 185; 186]. Based on medical imaging research in music training, researchers successfully developed interventions to enhance executive functioning and working memory in older adults [187]. Similarly, imaging findings in reading-related brain activities made it possible to design interventions to improve reading skills over time in dyslexic children [188; 189]. Surveyed educators largely believe understanding the brain is important to the design and delivery of teaching [43]. Berninger and Winn found that integration of neuroscience and learning science may promote educational evolution [30]. Dahlin *et al.* found that training can transfer between two tasks that engage overlapping processing components and brain regions [190]. Specifically, the neuroimaging findings of the role of working memory in prose writing [191; 192] have led to a series of instructional intervention studies showing writing problems can be improved [131; 132; 133]. Inspired by research in Psychology and other domains using medical imaging, we believe similar benefits for code writing may be available.

Beyond medical imaging, Parnin proposed a model focused on how a programmer manages task memory, specifically during multi-tasking and interruptions [193]. Of the previous studies combining neuroimaging or cognitive neuroscience with software engineering, as of 2021, none has investigated the effect of data structures on brain activity or explicitly investigated the relationship between data structures and spatial ability. In addition, no previous study has compared fMRI to fNIRS in the domain of software engineering.

Between 2020 to 2021, software engineering community has benefited from the uses of eye trackers. The results of eye-tracking studies add to the existing body of knowledge on how developers perform different software engineering tasks and how they use different models and representations along with source code to understand software systems. However, eye trackers are not without shortcomings and unlike neuroimaging, they do not provide insight into the brain activities. As a result, in a handful of studies, researchers started to use eye tracking simultaneously with electroencephalography (EEG) [181], fNIRS [38] and fMRI [176]. To the best of our knowledge, only Peitek *et al.* [176] performed a conjoint study to simultaneously use eye tracking and fMRI while providing a comprehensive analysis of the combined data.

In this chapter, we presented the background of psycho-physiological measures that are used in this thesis, including fMRI, fNIRS and eye tracking. We also introduced relevant software engineering and psychology tasks, as well as related work, that are involved in this thesis. Starting from the next chapter, we will present the three research components introduced in Chapter 1 that form the main body of this thesis to investigate and understand users' cognition in software engineering tasks. In the next chapter, we will first investigate the cognitive process in data structure manipulation, one of the most fundamental activities in software engineering.

CHAPTER 3

Representation of Data Structure Manipulation

In this thesis, we investigate the cognitive processes of data structure manipulations, code writing and code review, to provide a more complete picture of user cognition in software engineering activities. In this chapter, we begin by focusing on understanding the cognitive processes of data structure manipulation, one of the most fundamental activities in programming and software engineering, with a comparison to spatial ability. As the first chapter presenting studies adapting medical imaging in software engineering research, we also present a comparison between the fMRI and fNIRS paradigms, and discuss experimental guidelines for future research combining medical imaging and software engineering.

Data structures are a fundamental element in computer science that affect the performance and cost of many systems [194; 195; 196; 197]. Data structure choice and usage influence many aspects of software engineering, including maintainability [198], fault tolerance [199], reliability [200], and scalability [201]. Despite the importance of data structures in software development, we have a limited understanding of the subjective cognitive processes underlying their employment.

3.1 Overview of Experimental Design, Results and Contributions

In this thesis, we leverage two key insights to decode the neural representations of several classes of data structures and their manipulation[1]. First, we investigate the relationship between data structures and spatial ability. Spatial ability, which is first introduced in Chapter 2, is often measured via *mental rotation* tasks like the one illustrated in Figure 3.1 [116; 120; 115; 202]. Second, we use both fMRI and fNIRS, the two popular medical imaging techniques introduced in Chapter 2, to provide objective measurements of active brain function and establish a grounded understanding of mental processes associated with data structure manipulation. By comparing these neuroimaging modalities, we develop best practices for imaging investigations of software engineering. To the best of our knowledge, only one previous study focused on the relationship between software engineering tasks and spatial ability [10]. This previous work relied on interviews with students to understand their thought processes and suggested that programmers use visual representations to reduce the level of abstraction of data structures. However, no quantitative relationship has been investigated. Drawing inspiration from previous work, we consider spatial ability in the context of data structures to be the capacity to mentally represent, remember and manipulate spatial relations between elements of data.

We conducted a human study in which 76 participants mentally manipulated lists, arrays, and trees. Participants also completed mental rotation tasks involving the ability to determine if two perspective drawings portray the same three-dimensional shapes. In our study, we use mental rotation tasks to provide a solid neurological basis for spatial ability against which the cognitive processes associated with data structure manipulation can be compared.

The contributions of the work presented in this chapter are as follows:

- We report on a human study involving 76 participants and two medical imaging techniques, the largest such study we are aware of for software engineering.

[1] In this thesis, we focus on exploring the cognitive process of tasks on the neurological level.

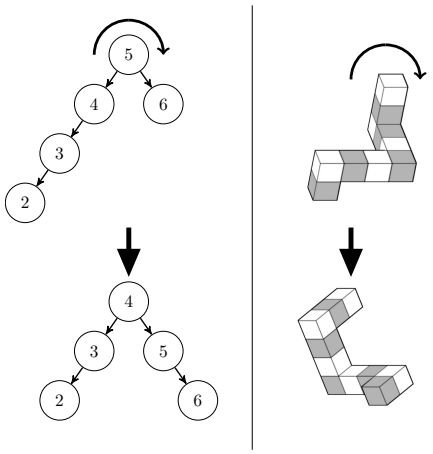


Figure 3.1: Illustration of the investigation on data structure manipulation and spatial ability: on the left, an unbalanced binary tree is rotated about node 1 to produce the tree on the bottom left. On the right, a three-dimensional object is rotated in space as shown in the bottom right. We investigate how the brain represents these two activities using medical imaging techniques.

- We find that data structure and spatial operations are related but distinct neural tasks: they use the same focal regions of the brain but to different degrees.

- We demonstrate that problem difficulty matters at a neural level in computer science, with more complex stimuli inducing a relatively higher cognitive load in data structure tasks than in mental rotation.

- We find that fMRI and fNIRS measurements broadly agree for the claims in this study. However, fNIRS cannot distinguish some activities as clearly as can fMRI. On the other hand, fMRI may influence participant accuracy. Care is needed when using medical imaging for software engineering.

- We present evidence from a qualitative investigation showing that imaging can find connections that subjective self-perceptions may overlook.

This chapter contributes to a fundamental understanding of cognitive processes in software engineering including (1) studying data structures with neuroimaging, (2) studying the relationship between data structure manipulation and spatial ability, and (3) comparing fMRI and fNIRS in the context of software engineering.

3.2 Experimental Setup And Methods

Having provided an overview of medical imaging and mental rotation, we present our study protocol to decode the neurological bases of data structures and their relationship with spatial ability and difficulty. Materials (*e.g.*, all stimuli and de-identified data) are available at the project's website.[2]

3.2.1 Overview

In this human study, participants completed three blocks of tasks while being scanned by either fMRI or fNIRS. Stimuli consisted of data structure (*i.e.*, list, array, tree) and mental rotation tasks with varying levels of difficulty. This setup permits the controlled investigation of the relationship between data structures and spatial ability through the lens of difficulty and the choice of medical imaging modality.

3.2.2 Recruitment

We recruited 76 students from University of Michigan for this study. Email solicitations were made to a graduate student list as well as brief presentations in four upper-level undergraduate CS classes. Monetary compensation was offered. After standard filtering (see Section 3.2.3), the final pool contained measurements from 30 fMRI participants and 40 fNIRS participants. Prior to each experiment, participants were screened for the requisite computing background. Table 3.1 summarizes the demographic information for all participants. The protocol was approved by University of Michigan's Institutional Review Board.

3.2.3 Data Collection

Each participant completed the experiment in a single session. Upon arriving, they provided informed consent and completed a background questionnaire. After watching a training video, participants were prepared for scanning and began the task activities. Participants completed three

[2] https://web.eecs.umich.edu/~weimerw/fmri.html

Table 3.1: Demographic data of eligible participants in the study of data structure manipulation

Demographic Variables		# fMRI	# fNIRS
Sex	Male	16	30
	Female	14	10
Degree Pursuing	Undergraduate	23	31
	Graduate	7	9

task blocks of 30 stimuli each (90 stimuli in total). All stimuli were presented for up to 30s and required an A or B response. A red fixation cross, a mark used to center participants' gaze, was shown before each stimulus for 2s–10s. Both fMRI and fNIRS experiments used the same set of 90 stimuli.

Stimuli were subdivided into three categories: (1) lists and arrays (collectively referred to as "sequences"), (2) trees, and (3) mental rotation. Each task block consisted of 10 stimuli from each category. The stimuli order was chosen randomly per participant. Participants were directed to respond as quickly and accurately as possible. After the scanning, participants completed a post survey to provide verbal explanations of their choices and actions.

Our experimental task protocol was designed to accommodate both fMRI and fNIRS. For the fMRI experiments, participants lay in an fMRI machine (see Section 2.1.1.1) holding MR-compatible buttons and remained in the machine for the entire scan (see Figure 3.2). In contrast, fNIRS participants sat in a chair wearing an fNIRS device (see Section 2.1.1.2) using a standard keyboard and monitor (see Figure 3.3). Participants were asked to remain still, but were permitted five minute breaks between each task block. As mentioned in 3.2.2, data from 6 individuals were removed due to difficulties presented when collecting fMRI data (e.g., discomfort in the machine, incomplete dataset, or excessive head motion). In the fNIRS analyses, data from all 40 individuals could be used.[3]

We now provide technical details suitable for conducting or replicating similar research. Section 3.2.4 continues with a discussion of the stimuli used in our experiment.

[3]Although no fNIRS data were removed due to noise, fNIRS does rely on differences in the absorption of near-infrared light, which can be obstructed depending on properties of a participant's hair such as color and thickness.

Figure 3.2: fMRI machine used in the experiment. The participant lies flat in the center of the bore.

3.2.3.1 fMRI Acquisition

In this experiment, we used fMRI to collect high-resolution imaging data following best practices from neuroimaging [203; 204]. All imaging procedures were conducted on a 3T General Electric MR750 with a 32-channel head coil at *University of Michigan Functional MRI Laboratory*. High-resolution anatomical images were acquired with a T_1-weighted spoiled gradient recall (SPGR) sequence ($TR = 2300.80$ ms, $TE = 24$ ms, $TI = 975$ ms, $FA = 8°$; 208 slices, 1 mm thickness). Prior to the functional scans, we obtained estimates of the static magnetic field using spin-echo fieldmap sequences ($TR = 7400$ ms, $TE = 80$ ms; 2.4 mm slice thickness). Functional MRI data were then acquired during both a resting state and during three task-related runs. All scans employed a $T_2{}^*$-weighted multiband echo planar imaging sequence sensitive to the BOLD contrast ($TR = 800$ ms, $TE = 30$ ms, $FA = 52°$; acceleration factor $= 6$), with whole-brain coverage over 60 slices (2.4 mm^3 isotropic voxels).

3.2.3.2 fNIRS Acquisition

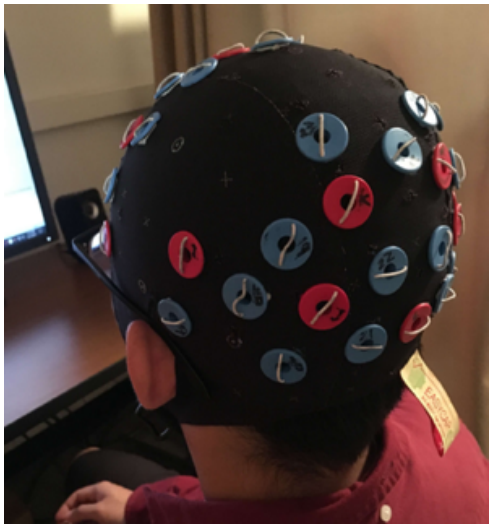
In this experiment, we collected data using the TechEn Inc. CW6 fNIRS system with an above-average number of light detection channels, allowing for a broader view of the brain activities than many published fNIRS studies (cf. [173; 174]). This system contains two laser diodes at 690 nm and 830 nm with fiber optic cables to transmit light between the instrument and a sensor probe on the participant's head. We designed three head caps to accommodate different head sizes (head circumference: 58 cm, 60 cm, 62 cm) based on the international 10–20 system [205; 206; 207] (see Figure 3.3a). For registration of the fNIRS cap [206], the cap center was aligned with the 10–20 point fPZ (above the bridge of the nose, see [207] for more measurement details). The cap included 16 light emitters and 32 detectors, spaced 3 cm apart, yielding 61 data collection channels[4] deployed at different regions. Regions were chosen based on previous neuroimaging studies of program comprehension and mental rotation [35; 119], and consisted of 15 Brodmann areas (see more details in Chapter 2.1.1). Signals were sampled at 50 Hz and then resampled to 2 Hz for analysis.

3.2.4 Materials and Design

As described in Section 3.2.3, participants were presented with three categories of stimuli: (1) sequences, (2) trees, and (3) mental rotation. Each stimulus from the first and second categories included a starting data structure, an operation to perform, and two answer choices (Figure **??**). Answers were either numerical values to describe the outcome of an operation or candidate data structures resulting from an operation. A sequence appeared as either a linked list or an array. For simplicity of modeling, we defined the *difficulty* of a sequence or tree task to be the total number of elements present — the N in Big-Oh notation.

The sequence tasks include merge, insert and swap operations. The tree tasks include binary search tree (BST) rotation, insertion and traversal operations. In mental rotation tasks, participants

[4] In theory, each emitter-detector pair could form a channel. In practice, our fNIRS hardware throughput limited us to 61 channels.



(a) fNIRS cap

(b) fNIRS environment

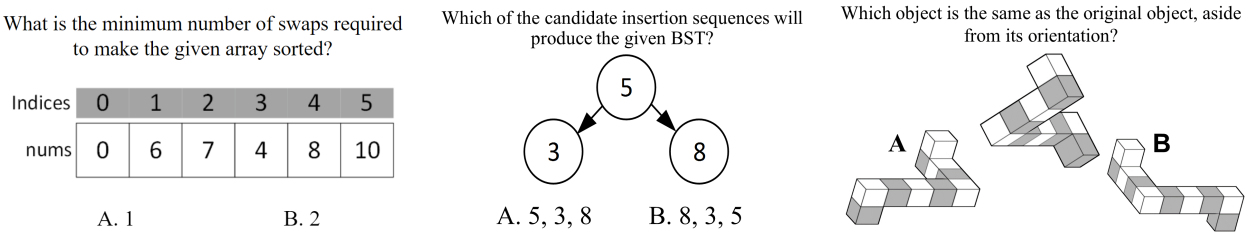
Figure 3.3: The self-made fNIRS cap: fitting on the head of a participant providing coverage of Brodmann areas 6–9, 17–19, 21, 39, 40, 41, 44–47 is shown on the left. On the right, a participant is shown completing the tasks in the fNIRS experimental environment.

were shown a starting three-dimensional object and two candidate objects. Participants chose the candidate that could result from a rigid rotation of the original (Figure 3.4c). The mental rotation stimuli were adapted from the Mental Rotation Stimulus Library established by Peters and Battista [208] with rotational angle difficulty. Figure **??** shows simplified examples. Stimuli are available at the project's website[5].

In the fMRI experiment, the stimuli were presented as images on a screen in the back of the scanner. Participants viewed stimuli via a mirror mounted atop the head coil.[6] Conversely, in the fNIRS experiment, the stimuli were presented as images on a computer monitor next to the fNIRS device (Figure 3.3b).

[5]https://web.eecs.umich.edu/ weimerw/fmri.html

[6]A helmet-like casing that surrounds the head and is essential for capturing high-quality images [209].



(a) Sequence (List or Array) (b) Tree (c) Mental Rotation

Figure 3.4: Example task stimuli: Sequence, Tree and Mental Rotation. The examples for Sequence and Tree stimuli are simplified for presentation space and clarity.

3.3 Approach

In this section, we present details on the mathematical analyses applied to fMRI and fNIRS data. Our goal is to localize brain activations from task-related changes in the BOLD response (fMRI) or light absorption (fNIRS). Such analyses pose complicated statistical challenges, involving the interpretation of *hemodynamic* responses across anatomically and functionally diverse participants, which themselves are indirect metabolic proxies for underlying *neuronal* (i.e., molecular/cellular) responses. We used standard preprocessing techniques to identify and remove artifacts, validate model assumptions, and standardize locations of brain regions across participants. We then used general linear models to obtain estimates of task-related brain activations within voxels (fMRI) or channels (fNIRS) based on the canonical hemodynamic response function. Finally, we performed statistical tests at both individual and group levels to test for significant brain activations, including subsequent correction for false positives.

Notation. As described in Chapter 2.1.1, we use the neuroimaging notation $\mathsf{A} > \mathsf{B}$ to refer to the *contrast* (or difference) between two task conditions. For example, $\mathsf{Sequence} > \mathsf{Tree}$ refers to the comparison of brain activations during sequence vs. tree manipulation. Contrasts are *directional* tests: the aforementioned $\mathsf{Sequence} > \mathsf{Tree}$ contrast will specifically attempt to identify regions in which average sequence task activity is *greater* than tree manipulation.

3.3.1 fMRI Analysis Approach

Preprocessing. A critical first step in the analysis of fMRI data is *preprocessing*, which serves to correct systematic sources of noise and transform individual brains into a standard space for cross-participant comparison. We employed a number of standard preprocessing procedures using the Statistical Parametric Mapping 12 (SPM12, Wellcome Trust Centre for Neuroimaging, London) software in Matlab. First, we computed *voxel displacement maps* (VDMs) using images from the fieldmap sequence. We then realigned the functional scans after accounting for head motion over time; the VDMs were used to "unwarp" geometric distortions from motion. Next, the anatomical scans were segmented, skull-stripped, and spatially coregistered to the functional data. All images were then transformed into a standard space according to the Montreal Neurological Institute (MNI152) template [210]. Finally, we computed a brain mask using gray and white matter segments of the anatomical scans — this was applied in subsequent statistical analyses to prevent identification of false positive signals within ventricles or outside of brainspace.

First-level analysis. Functional MRI analyses are *multi-level*. First-level models are specified on individual participant data — the results are then combined in a group-level model to assess average task-related changes in brain activity. We specified two first-level general linear models (GLMs) per participant. Briefly, these analyses require us to *predict* the BOLD response to each condition — voxels whose timeseries align with the predicted response are "task-sensitive". In each GLM, we specified regressors for Sequence, Tree, and Mental stimuli across all runs. The duration of each event was curtailed to participant response times. These were convolved with the canonical hemodynamic response function (HRF) and high-pass filtered ($\sigma = 128$ s) to remove low-frequency noise. In one model, we additionally specified a *parametric modulator* for each condition to determine whether the magnitude of the BOLD response scaled linearly with trial difficulty. All models were fit using robust weighted least squares (rWLS) [211], which first obtains estimates of the error variance at each timepoint and reweights the images by a factor of 1/variance to reduce the influence of noisy scans (e.g., due to head motion). This procedure homogenizes the residual timeseries and obtains optimal parameter estimates for each condition.

Contrasts and group-level analysis. Following first-level model estimation, we computed pairwise contrasts to determine mean differences in activity between conditions. These were estimated on a within-participant basis (i.e., on first-level models). We applied a 5 mm^3 full-width at half maximum (FWHM) Gaussian smoothing kernel to each contrast map and carried them upward into group-level *random effects* analyses. A GLM in this context allows us to assess average activity across *all* participants, accounting for inter-individual variance to make some population-level inference. The end result is a *statistical parametric map* of t-values describing clusters of significant activity for a given task-related comparison. Importantly, all models and tests described here were done *voxelwise* — that is, a GLM was specified and estimated for each of nearly 73,000 voxels in brainspace. We therefore applied a *false discovery rate* (FDR) threshold at $q < .05$ to control for false positives as a result of multiple comparisons.

3.3.2 fNIRS Analysis Approach

Preprocessing. The raw fNIRS data are light signals transmitted through the channels between emitters and adjacent detectors on the fNIRS cap. The light signals were converted to a measure of the optical density[7] change over time that results from hemodynamic responses.

First-level analysis. Statistical analyses for fNIRS follow the same general principles as fMRI. We specified within-subject, first-level GLMs to model fNIRS optical density measurements in all the channels that were statistically related to the timing of the hemodynamic responses (as determined by convolving timeseries of stimulus events with the canonical HRF). In fNIRS, systemic physiology and motion-induced artifacts are major sources of noise and false positives. We therefore fit our models using autoregressive-whitened robust regression [212], which controls for such confounds and affords optimal parameter estimation. Then, we applied t-tests to the regression coefficients describing the task-related brain activations modeled for every participant. We additionally separated tasks into three difficulty levels and constructed GLMs to analyze the effect of task difficulty on neural activity.

[7]The degree to which a refractive medium retards transmitted rays of light.

Contrasts and group-level analysis. As with the fMRI analysis, we computed pairwise contrasts to determine mean differences in activity between conditions, estimated on a within-participant basis. Next, we conducted a group-level analysis to summarize the first-level regression coefficients. A mixed effects model was used to examine the average group-level response, with individual participants treated as random effects. Finally, we applied an FDR threshold at $q < .05$ to control for false positives from multiple comparisons.

3.4 Results and Analysis

We present quantitative and qualitative analyses to address the following research questions:

RQ 3.1 Do data structure manipulations involve spatial ability?

RQ 3.2 What is the role of task difficulty?

RQ 3.3 Do fMRI and fNIRS agree for software engineering?

RQ 3.4 How do self-reporting and neuroimaging compare?

For simplicity of presentation, we use Code to refer to sequence (array and list) and tree tasks collectively.

3.4.1 RQ 3.1 (Data Structures & Spatial Ability) — fMRI

We began with a broad examination of mental rotation vs. code tasks, independent of task difficulty: this would allow us to determine whether there were reliable differences between mental rotation and the two data structure tasks on average. A group-level test of Code $>$ Mental yielded no significant activations after FDR thresholding (i.e., no regions showed consistently stronger activations across both tree and sequence tasks relative to mental rotation). However, Mental $>$ Code revealed robust increases in activation (FDR-corrected) of several regions commonly associated

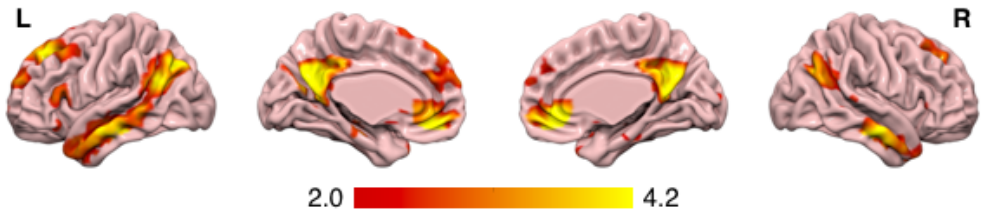


Figure 3.5: Weight map: significant cluster of brain activity for Mental $>$ Tree, independent of task difficulty. "Hotter" colors indicate regions showing a larger magnitude difference between the two tasks (i.e., more activity during mental rotation relative to tree manipulation).

with the brain's "default mode network" (DMN) [213]. Most notably, we observed bilateral recruitment of wide swaths of posterior cingulate cortex (PCC; BA 31) and medial prefrontal cortex (mPFC; BA 8), including subgenual anterior cingulate cortex (sgACC; BA 32). On the lateral face, there emerged a large cluster of activity in the left angular gyrus (AG) / temporoparietal junction (TPJ) (BA 39, 21–22), with additional clusters extending rostrally along the superior temporal sulcus (pSTS) and middle temporal gyrus (MTG) to the temporal pole (BA 21, 38). These anterior temporal cortex clusters were also largely bilateral. The DMN is heavily implicated in various types of *mental simulation*, as required by the tasks performed here.

Given that mental rotation reliably activated DMN regions more than the two code tasks, we applied more focal contrasts to determine whether there were specific differences between Mental $>$ Tree and Mental $>$ Sequence. This revealed that the Mental $>$ Code effect was primarily driven by Mental $>$ Tree (Figure 3.5). While Mental $>$ Sequence yielded significant differential activations in midline DMN regions such as the PCC and mPFC, these clusters had relatively minimal spatial extent. Patterns of activity related to Mental $>$ Tree, however, were nearly identical to those observed in the comprehensive Mental $>$ Code contrast (Pearson's $r = 0.97, p < .001$). As with the omnibus Code $>$ Mental contrast above, the inverse contrasts (Tree $>$ Mental and Sequence $>$ Mental) also had no voxels survive FDR thresholding.

fMRI results suggest that there are more similarities than differences during mental rotation vs. software engineering tasks. A number of DMN regions involved in mental simulation were recruited more heavily during mental rotation; nevertheless, 95% of voxels were statistically indistinguishable between Mental and Tree tasks.

3.4.2 RQ 3.1 (Data Structures & Spatial Ability) — fNIRS

Table 3.2: Summary of fNIRS results. Each column corresponds to a particular task. Each row corresponds to a particular Brodmann Area used during that task along with the range of t-values measured by all fNIRS channels on that BA. Positive t-values indicate stronger activation while negative t-values indicate weaker activation. We report all t-values with $p < 0.01$: all reported results are significant.

Sequence		Mental		Tree		Sequence > Mental		Mental > Tree		Sequence > Tree	
BA	t-value range	BA	t-value range	BA	t-value range	BA	t-value range	BA	t-value range	BA	t-value range
6	2.5 – 5.0	6	2.8 – 4.3	6	3.8 – 4.6					6	2.7 – 2.7
7	4.7 – 5.5	7	5.9 – 6.4	7	5.1 – 7.2						
8	2.6 – 5.1	8	2.9 – 5.5	8	2.5 – 5.6						
9	2.6 – 5.1	9	5.5 – 5.5	9	2.7 – 5.3						
17	3.1 – 4.9	17	3.2 – 6.2	17	2.6 – 5.3	17	-2.4 – -2.4				
18	3.8 – 5.2	18	5.3 – 6.9	18	4.2 – 5.3	18	-2.4 – -2.4	18	2.6 – 2.6		
19	4.0 – 6.6	19	5.3 – 9.1	19	4.2 – 7.3	19	-4.3 – -3.2	19	2.4 – 4.3		
39	3.7 – 7.1	39	4.1 – 7.9	39	4.4 – 7.9	39	-3.3 – -3.3	39	2.4 – 2.4		
						41	-2.3 – -2.3				
						44	-3.3 – -2.6	44	2.6 – 3.4		
						45	-5.0 – -2.4	45	3.5 – 3.5		
46	3.8 – 4.1	46	3.5 – 4.6	46	4.7 – 5.6	46	-5.9 – -2.4	46	2.7 – 4.3	46	-2.6 – -2.6
						47	-5.9 – -5.0	47	3.4 – 4.3		

Table 3.2 summarizes the fNIRS results. We first examined brain activations comparing each task to a rest condition. The columns Sequence, Mental and Tree show the Brodmann Areas that are significantly activated during the task categories ($p < 0.01$ and $q < 0.05$). The t-values range from 8 (much stronger activation) to -8 (much weaker). We observe that the three categories of tasks all involve significant activations in exactly the same brain regions: BA 6–9, 17–19, 39 and 46.

In the frontal lobe, the premotor cortex and supplementary motor cortex (BA 6), and the frontal eye field (BA 8) showed activation. In the parietal lobe, the part which is associated with visuo-motor coordination presented activation (BA 7) and part of Wernicke's area showed activation (BA 39). We also observed strong activation in the primary, secondary and associative visual cortex (BA 17–19). Finally, regions of the dorsolateral prefrontal cortex (BA 9, 46) showed activations for all tasks. All the brain areas listed in the table passed FDR correction ($q < 0.05$).

Having established a broad similarity in how the three tasks each differ from a rest state, we narrowed the investigation by examining how the tasks differ from each other. In Table 3.2, the column Sequence $>$ Mental shows the brain activation results when comparing sequence tasks and mental rotation tasks. Areas related to vision (BA 17–19), Wernicke's area (BA 39) and the prefrontal cortext (BA 46) showed very different patterns of activation between the data structure task and mental rotation. In addition, areas related to language processing (BA 41, 44–45, and 47, which include Broca's Area) strongly distinguish the two. As we observe here, an area (e.g., BA 41) may not significantly distinguish Sequence from a rest state or Mental from a rest state, but may significantly distinguish them *from each other*.

However, the Mental $>$ Tree and Sequence $>$ Tree distinctions are far less compelling. In a comparison, t-values near to either 8 or -8 are relevant. While Sequence $>$ Mental features three areas that reach a magnitude of 5 or more, the other two contrasts never reach a magnitude of 5 and involve fewer regions and channels. In an fNIRS analysis [214; 215], contrasts of that strength result in a conclusion that Mental and Tree, as well as Sequence and Tree, are similar tasks.

fNIRS results demonstrate that mental rotation and data structure tasks involve activations to the same brain regions. However, while Sequence $>$ Mental may be a compelling contrast, the fNIRS evidence does not support the claim that the other tasks are distinct.

3.4.3 RQ 3.2 (Task Difficulty) — fMRI

When we considered the difficulty of each task, we found a significant effect in Sequence $>$ Mental (Figure 3.6). Larger sequence tasks elicited stronger activations across a wide extent of the brain (FDR-corrected). With the exception of PCC, there was little to no overlap with DMN regions (as seen in the contrasts in 3.4.1). Rather, the largest clusters included bilateral postcentral gyrus (BA 40), left inferior frontal gyrus (IFG; BA 44–45), bilateral dorsomedial PFC (dmPFC; BA 6, 8), bilateral anterior insula (BA 13), and bilateral ventral precuneus extending into visual association cortex (BA 18). The heavy recruitment of frontoparietal regions — particularly in the left hemisphere — suggests an increase in cognitive load [216] scaling with the total size of the stimuli.

That is, we found that the brain works measurably "harder" (i.e., there is a larger magnitude BOLD response) for more difficult problems. Because the relationship between mental rotation difficulty and the BOLD signal is so well-established in psychology and cognitive neuroscience [120; 121], it is particularly compelling that we observe a significantly larger effect (in terms of cognitive load and top-down control rising with more complex stimuli) for sequence data structures in software engineering than for mental rotation.

A similar analysis with our fNIRS data revealed no significant findings for the effect of task difficulty on neural activity. This is likely due to fNIRS lacking the penetrative depth and spatial resolution of fMRI.

The brain works measurably harder for more difficult software engineering problems (in terms of cognitive load). Moreover, the regions activated suggest a greater need for effortful, top-

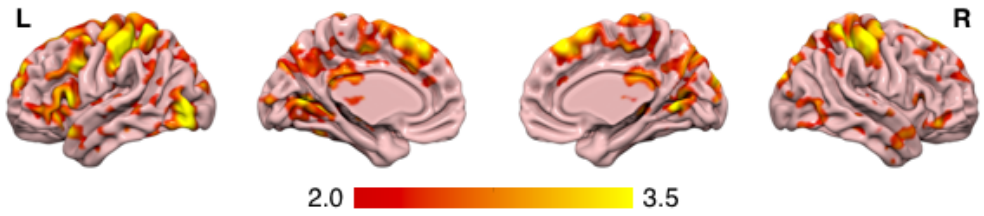


Figure 3.6: Weight map: significant clusters of brain activity for Sequence $>$ Mental, accounting for task difficulty. "Hotter" colors indicate regions showing a larger magnitude difference between the two tasks (i.e., more activity during difficult sequence manipulation trials relative to difficult mental rotation trials).

down cognitive control when completing challenging sequence manipulation tasks.

3.4.4 RQ 3.3 (fMRI and fNIRS Agreement)

Our fMRI and fNIRS measurements and analyses both support the claim that mental rotation and data structure tasks differentially recruit a number of brain regions. However, while fMRI evidence supports a very robust Mental $>$ Tree contrast, the fNIRS evidence is insufficient to support that same claim. This is sensible when we consider the regions yielding the largest differences in fMRI: they largely correspond to structures (e.g., the medial prefrontal cortex and posterior cingulate) that fNIRS cannot measure. Very informally, the parts of the brain that distinguish mental rotation from tree manipulations are too far "inside the skull" for fNIRS to see: its near-infrared light cannot penetrate deeply beyond regions near the cortical surface.

However, while fMRI is more spatially-resolved, its restrictive and alien environment can also be more daunting for participants. We compared participant performance (i.e., whether or not they gave the correct answer and how long it took) for fMRI and fNIRS; such information was available for 30 fMRI and 40 fNIRS participants. Recall that the questions were identical and the participants were drawn from the same pool. The average accuracy of fNIRS participants, 92%, was significantly higher than the 85% accuracy of fMRI participants ($t = 4.50$, $p < 0.01$) with no

significant difference in response time ($t = 0.70$, $p = 0.25$). This could be a very relevant concern for medical imaging studies of productivity, expertise, accuracy or similar software engineering issues.

> fMRI and fNIRS agreed that many areas similarly activate during data structure and mental rotation tasks. However, there were also differences between the tasks that fNIRS was not able to observe. In addition, the fMRI environment had a significant effect on participant accuracy.

3.4.5 RQ 3.4 (Self-Reporting & Neuroimaging) — Qualitative

We also conducted a qualitative analysis of survey data focusing on the correlation between explanations provided by participant and neuroimaging data. Data was available for 72 of our 76 participants. At a high level, we find that self-reporting often subtly contrasts with analyses from fMRI and fNIRS data. Complete (de-identified) survey information is available with our other experimental materials and scanned measurements; for reasons of space we focus here on a single indicative question.

Participants were asked to compare and contrast a mental rotation task with an BST rotation task. Of the 72 responses, 70% reported *no similarity* between the two tasks. The following quote is indicative: "I don't think those two kinds of tasks were similar. Tree rotation was an idea acquired from CS classroom [sic], but mental rotation was an action more natural to me and easier to perform." However, this subjective experience does not align with measured observation that the same brain regions are recruited to solve both tasks. Even if mental rotation and tree rotation feel subjectively different, changes to brain regions and brain region connectivity have been shown to correlate with learning rates and expertise [217; 218]. It may be, for example, that exercises related to spatial ability can help improve student performance on certain data structure tasks (e.g., because mastering one changes a brain region recruited by the other). While speculative, this is simply one example of a research avenue that is encouraged by medical imaging data but entirely hidden if only self-reporting data is used.

These findings reinforce a considerable body of work on unreliable self-reporting (both in psychology [219; 220] and in computer science, including fields such as security [16], human-computer interaction [17], and software maintenance [18]). As previous studies have relied on self-reporting to study mental processes associated with data structures [10; 134], this evidence informs future research of the importance of neuroimaging (or similar techniques) when studying the cognitive processes underlying software engineering tasks.

> While medical imaging data found a nuanced relationship between mental rotation and data structure tasks, including the involvement of the same brain regions, subjective self-report only rarely mentioned any connections.

3.5 Threats to Validity

In this section, we describe threats to internal and external validity in this experiment.

One potential threat to internal validity concerns whether or not our data structure tasks measure what they claim to be measuring (i.e., data structure manipulation). The thought processes that participants used when answering may not be identical: indeed, there is significant inter-participant variance in the neural representation of this problem solving. In addition, the particular data structures and tasks we chose are not representative of all of software engineering (e.g., skip lists, tries, heaps, maps, etc. are not considered). While we mitigate this somewhat by considering fundamental structures (linear sequential structures and branching trees), it is important not to generalize our results far beyond what was directly measured.

Our use of mental rotation tasks as a baseline for spatial ability is one potential threat to external validity, as mental rotation and data structure manipulations differ in their *rigidity*. In spatial ability tasks, rigid transformations are those where distances between every pair of points on an object is preserved [221]. When studying the relationship between data structure manipulation and spatial ability, operations such as insertion, tree rotation, and merging may be more amenable to

comparison with non-rigid transformations. However, we believe that mental rotation serves as a useful baseline (see Section 2.2.1) for relating data structures to spatial ability. Mental rotation is a paradigm case of spatial ability, and has been classified on the basis of difficulty both with and without medical imaging [117; 118; 119].

The data structure stimuli used in this experiment may pose an additional threat to external validity. Due to the inherent limitations of fMRI and fNIRS (see Section 2.1.1.3), we explicitly used stimuli that took no longer than 30 seconds to finish. Thus, by focusing only on relatively short data structure tasks, our results may not generalize to real-world software engineering tasks. We mitigate this threat slightly by choosing stimuli from college-level courses, which commonly focus on associated fundamental skills. However, this emphasis on tasks that are much shorter than many of those performed by practicing software developers is a significant limitation of the current use of medical imaging techniques in software engineering [35; 67; 173; 174; 38; 37; 36; 175; 176].

A final threat to external validity is the pool from which we selected participants. By only recruiting undergraduate and graduate students, our results may only generalize to those with university-level programming experience and education.

3.6 Costs, fMRI, fNIRS, and Research

Medical imaging studies, while still quite rare, are becoming more common in the software engineering literature [35; 67; 173; 174; 38; 37; 36; 175; 176]. fMRI and fNIRS present tradeoffs between cost, fidelity, experimental convenience, and experimental verisimilitude. In this section, we discuss their tangible and intangible costs, including those associated with participant recruitment, equipment cost and time.

As discussed in Section 2.1.1.3, fMRI poses significantly higher monetary costs than fNIRS. In our study, the cost of fMRI was $575/hour (including the equipment, the fMRI-provided technician, etc.); each participant required 30 minutes of preparation, up to 75 minutes of scanning, and the presence of two researchers. By contrast, in our institute, the use of fNIRS equipment was

free; each participant required 30 minutes of preparation time to fit the cap, up to 75 minutes of scanning, and the presence of two researchers.

In addition, each approach comes with recruitment restrictions. For example, fMRI typically requires corrected-to-normal vision (because of the mirror/projection setup) and is not approved for pregnant women or those with medical implants or head tattoos, etc. In some cases, participants may not be able to finish a fMRI scanning due to claustrophobia. On the other hand, fNIRS may place significant practical restrictions on the use of participants with dark, thick hair. In practice, we found the fNIRS restrictions to be less onerous (resulting in 0 unusable applicants compared to 4 for fMRI).

Software engineering researchers must carefully weigh the costs and benefits. At a high level, the two approaches provide broadly similar evidence. fNIRS requires the researcher to identify relevant brain areas in advance for cap construction (Section 2.1.1.3) and cannot penetrate some areas relevant to software engineering (Section 3.4.4). On the other hand, while fMRI is regarded as the gold standard for imaging accuracy, it cost roughly $20,000 more to acquire the fMRI data than the fNIRS data for this experiment, and the environmental constraints of fMRI may influence participant accuracy (Section 3.4.4). As a broad generalization, researchers investigating a computer science topic for the first time may favor fMRI; once the relevant brain areas have been identified, if those regions are accessible to fNIRS light, a more cost effective and ecologically-valid study can be conducted via fNIRS. If the proposed study requires more freedom of motion or a quiet environment, involves more than one participant (*e.g.*, pair programming, face-to-face communication, etc.), or uses metal equipment (*e.g.*, a tablet or cellphone), fMRI is not an option without significant extra work.

3.7 Chapter Summary

We investigated the neural representations of fundamental data structures and their manipulations. We hypothesized that data structures are related to spatial ability. Our two key insights were the

use of multiple medical imaging approaches and the use of the mental rotation paradigm to serve as a baseline for measuring spatial ability.

Our study involved 76 participants, at least two times larger than previous studies investigating software engineering with medical imaging and is the first to investigate the neural representations of data structures.

We found that data structure and spatial ability operations are related: both fMRI and fNIRS evidence demonstrates that they involve activations to the same brain regions (*e.g.*, Section 3.4.1 and Section 3.4.2, $p < 0.01$).

However, the similarity relationship is nuanced: spatial ability operations and tree operations admit a significant contrast and are characterized by differentiated activation magnitudes (*e.g.*, Section 3.4.1, $p < 0.001$).

Further, some regions relevant to data structures are not accessible to fNIRS: fNIRS lacked the penetrating power to uncover the full evidence reported by fMRI (Section 3.4.4) and was unable to distinguish between two distinct tasks.

We also found that difficulty matters for data structure tasks: more complicated stimuli result in greater neural activation, and thus an increase in cognitive load (Section 3.4.3).

While a neural relationship between between spatial ability and data structure manipulation may seem clear in retrospect, it was not obvious to our participants, 70% of whom reported no subjective experience of similarity (Section 3.4.5).

Since our direct comparison of fMRI and fNIRS is unique in software engineering, we elaborate on both measurement and performance issues (Section 3.4.4), as well as monetary, protocol and recruitment issues (Section 3.6).

Data structures are critical to many aspects of software engineering, but no previous work has quantitatively investigated their neurological underpinnings. The use of medical imaging to understand cognitive processes in software engineering is still very new: this study is exploratory rather than definitive. Indeed, from our perspective it encourages future investigations, such as the exploration of how a neurological link between data structures and spatial reasoning can best

inform pedagogy or training. It is our hope that our concrete analysis of a lower-cost medical imaging alternative, as well as our direct analysis of a computing activity beyond the realm of code comprehension, will encourage more researchers to investigate the cognitive aspects of software engineering.

Having concluded our investigation on data structure manipulation and the comparison between fMRI and fNIRS, we will study a higher level and more complex software engineering activity, code writing, in the next chapter.

CHAPTER 4

Comparing Code Writing and Prose Writing

In the last chapter, we investigated the cognitive processes of data structure manipulation, a fundamental activity in software engineering, with a comparison to spatial ability, and also provided guidelines to select medical imaging techniques. In this chapter, we study a more complex software engineering activity: code writing.

Writing code is a crucial activity in software engineering. Despite this increasing prevalence of software and demand for skilled programmers, as of 2021, most of research still rely on traditional survey instruments and self-reporting, rather than an understanding of fundamental human brain function, when developing methods to support, improve, teach and evaluate code writing and editing.

As introduced in Section 2.2.2, previous work simultaneously studying code and prose writing has focused on non-imaging uses of one to aid in the instruction of the other. More explicitly, research in storytelling has used programming as a means to improve children's prose writing [222], and vice versa [223; 224]. In either direction, researchers reported similarity in the processes of code and prose writing, such as their sequence, structure, and object-oriented nature [225]. However, these qualitative findings have not been substantiated by medical imaging.

In this thesis, we provide the first instantiation of studying cognitive processes with medical imaging in code writing and present findings. Based on the comparison between fMRI ad fNIRS in the previous chapter, we choose to use fMRI as the medical imaging modality due to its stronger penetration power.

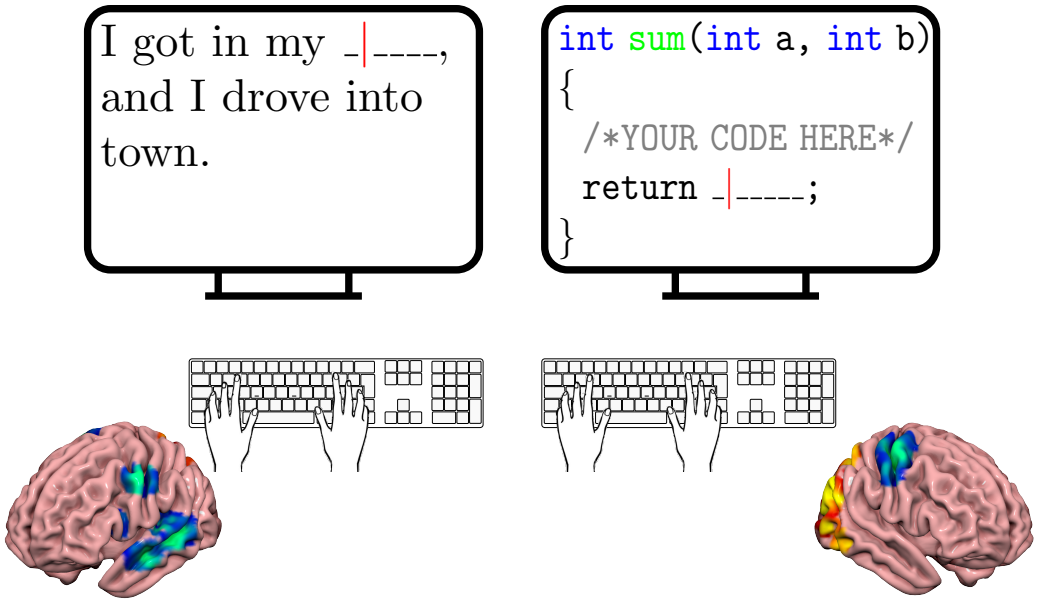


Figure 4.1: Illustration of the investigation on code and prose writing: we investigate the relationship between prose and code writing using functional brain imaging. Experimental controls systematically vary content (code vs. prose) and size (fill-in-the-blank vs. long response). Do code and prose writing exhibit the same patterns of neural activity?

There is a significant body of work studying the psychology of programming, ranging from the cognitive prerequisites of programming [33] to entire theories of the coding process [12], but this research has relied largely on observational evidence. Recent advances in medical imaging, particularly functional magnetic resonance imaging (fMRI), have improved researchers' ability to measure brain activity associated with various cognitive processes. As a non-invasive, *in vivo* technique, fMRI is an effective tool for clinical researchers studying brain function [76; 77; 78] and the effects of various treatments [226; 227; 228], as well as for psychology researchers mapping brain areas in activities as diverse as musical performance [28] and food cravings [29]. Findings using medical imaging have successfully transitioned to guiding behavioral and developmental improvement in domains like mathematics [42] and education [43].

While there have been fMRI studies of code *reading* (*e.g.*, [35; 67]) and *non-fMRI* studies of code writing (*e.g.*, [153; 142]), to the best of our knowledge there are no previous fMRI studies of code writing. We attribute this to two challenges: physics and design. First, normal keyboards cannot be safely placed or accurately read near magnetic resonance scanners. They interfere with

the fMRI measurements and the fMRI interferes with keyboard reporting. Second, imaging studies require carefully-controlled experiments, and no high-level design for a code writing contrast has been proposed (cf. Behroozi *et al.*'s contrast of whiteboard interview questions with pencil-and-paper versions [229], which changes the modality but uses identical tasks, or our approach in the previous chapter that contrasts data structure problems with mental rotation problems [15], which changes the task but not the modality). We combine two corresponding insights to overcome these challenges. First, we propose to employ a bespoke keyboard that moves all metal and control logic to a separate room. Second, we propose a two-by-two contrast setup: code vs. prose writing and fill-in-the-blank vs. long response (informally, single-word production vs. longer creativity).

Our use of prose writing as a baseline grounds our experiment and clarifies our results. Prose writing is a well-studied activity in psychology [124; 125; 126; 230; 231], and medical imaging has aided understanding of its underlying cognitive processes (see Chapter 2.2). For example, fMRI studies have provided insights into brain areas associated with prose writing [31] and the specificity of such regions across different prose writing tasks [129], in addition to addressing neural correlates of the roles of expertise [232] and creativity [127]. The contrast between code and prose writing in our experiment illuminates their differences and similarities at a neurological level.

4.1 Overview of Experimental Design, Results and Contributions

We conducted a human study in which 30 participants performed prose and code writing tasks in an fMRI scanner (Figure 5.1). Participants completed two types of tasks: fill-in-the-blank (FITB) and long response (LR). FITB tasks presented either a sentence or program containing a blank space, requiring the participant to provide the missing word or code snippet. In LR tasks, participants wrote prose or code from scratch to answer an open-ended question or meet a program specification.

Our primary finding is that code writing and prose writing feature significantly different patterns of neural activity, particularly in parts of the brain associated with attention control, working memory, and spatial cognition. While prose writing involves activation in canonical areas associated with language, code writing involves a very different set of right-lateralized regions associated with attention, memory, planning and spatial ability. Our experiment provides the first evidence of significant neural differences between *prose writing* (which is neurally similar to natural language) and *code writing* (which, we find, is *not*).

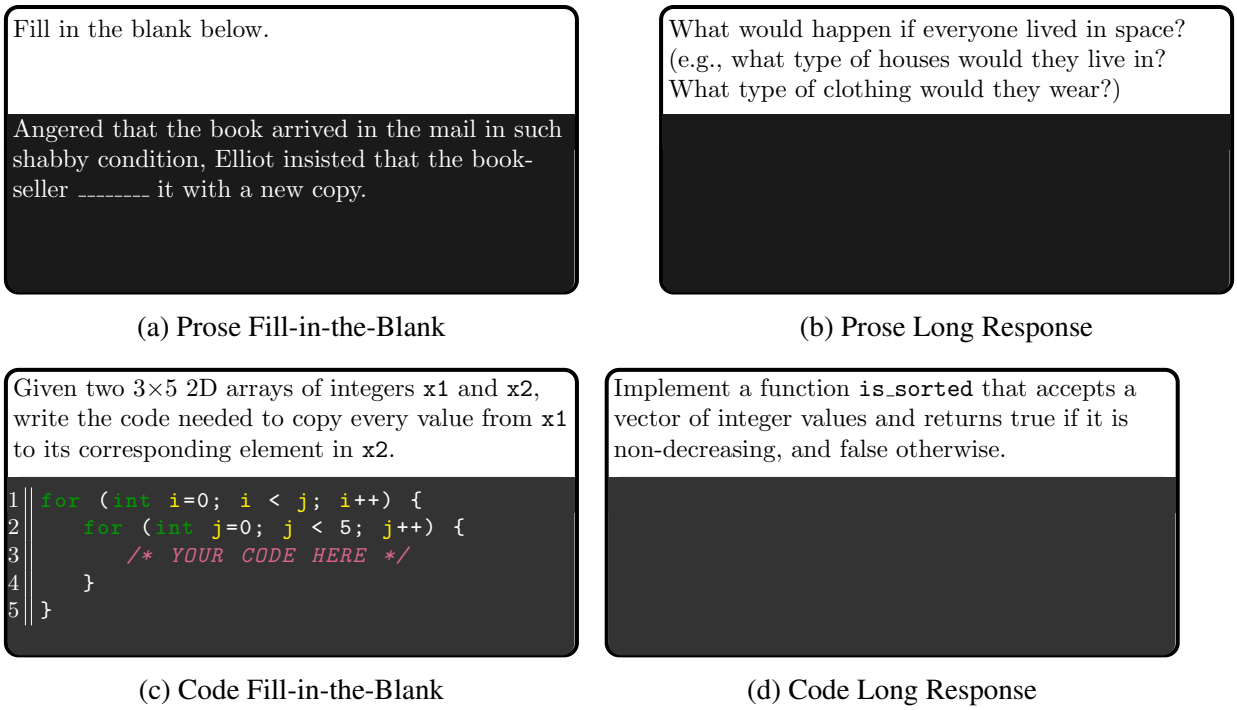
The contributions of this chapter are as follows:

- An fMRI study of 30 participants comparing code writing to prose writing. To the best of our knowledge, this is the first fMRI study to feature keyboard code writing. Our experimental design contrasts code, prose, fill-in-the-blank and long-response questions.

- A mathematical analysis of the results. After mitigating noise and correcting for false discovery rate ($q < 0.05$), we find that general code and prose writing feature distinct patterns of neural activity ($2.4 \leq t \leq 6.2$) related to attention, working memory and spatial cognition. For long-response writing questions, we find the clearest distinction we are aware of in the literature ($-7.0 \leq t \leq -3.1$ and $3.5 \leq t \leq 5.8$) between code (attention, memory, planning and spatial ability) and prose (language, letters and words).

- For replication and reproducible research, we make available our materials and methods on our project website.[1] These include our corpus of stimuli; our de-identified medical imaging data; our method for adapting a 101-key QWERTY USB keyboard for the fMRI environment; and a configurable program for stimuli presentation, editing and data collection.

[1] https://web.eecs.umich.edu/ weimerw/fmri.html



(a) Prose Fill-in-the-Blank

(b) Prose Long Response

(c) Code Fill-in-the-Blank

(d) Code Long Response

Figure 4.2: Example two-by-two task stimuli: code and prose writing. We investigated four categories of stimuli covering code and prose in fill-in-the-blank and long response scenarios.

4.2 Experimental Setup and Methods

We present a human study in which 30 participants underwent an fMRI scan while completing prose and code writing tasks. We discuss (1) the makeup and recruitment of our participant cohort, (2) how we developed our task materials, (3) the experimental protocol, (4) our method for collecting fMRI data, and (5) the construction of an fMRI-safe keyboard that enabled participants to freely write text and code during an fMRI scan.

4.2.1 Participant Demographics and Recruitment

We recruited 30 undergraduate and graduate computer science students at the University of Michigan. The protocol was approved by the University's IRB (HUM00138634). Table 4.1 summarizes demographic information for this cohort. Students who had completed coursework in data structures and who could safely undergo an MRI scan were eligible to participate. All participants

Table 4.1: Demographic data of the eligible participants in the study of code writing.

Demographic Variables	**# Participants**	
Sex	Male	20
	Female	10
Gender	Men	20
	Women	9
	Fluid	1
Degree Pursuing	Undergraduate	27
	Graduate	3

were native English speakers, right-handed, and had normal or corrected-to-normal vision. Each participant was offered a $75 cash incentive and a 3D model of their brain upon completion.

When participants elected to participate in the study, we collected basic demographic data (sex, gender, age, cumulative GPA, and years of experience) and socioeconomic status (SES) data. In addition, each participant completed three standard psychological measurement surveys: Positive and Negative Affect Scale (PANAS, emotional health), Autism Spectrum Disorder (ASD), and Need for Cognition (NFC, inclination for effortful cognition). Finally, we administered a short programming quiz to assess basic C/C++ programming skills.

Although we conducted a correlation analysis between these demographic and psychological measures and brain activities, none survived a strict false discovery rate correction ($q < 0.05$). We claim no significant demographic or attitudinal correlation with code or prose writing in our study. In the remainder of this paper, we thus treat our participants as a whole, rather than considering any subpopulation analyses.

4.2.2 Participant Tasks

Participants underwent an fMRI scan during which they completed a sequence of tasks associated with code and prose writing. Participants were shown a sequence of sentences or code snippets and asked to type a response while inside the MRI machine. We divided tasks into Fill-in-the-Blank (FITB) and Long-Response (LR) activities. In FITB, participants were shown a nearly-completed

sentence or code snippet and had 30 seconds to type a short word or expression that they thought best completed the sentence or snippet. In LR, participants had 60 seconds to write a complete response to a high-level task or question. Participant completed four categories of tasks, each lasting 20 minutes: (1) 17 FITB Prose tasks, (2) 9 LR Prose tasks, (3) 17 FITB Code tasks, and (4) 9 LR Code tasks. Examples of stimuli under each of these categories are shown in Figure 4.2.

Code Tasks We developed a corpus of code stimuli by adapting tasks from Turing's Craft [233], a library of short programming exercises used in web teaching evaluations [234], each with prompts and example correct solutions. For the FITB Code tasks, we selected a set of 17 prompt-answer pairs, and replaced a random portion of the solution with a blank line. Participants were asked to fill in that blank line. For the LR Code tasks, we selected a set of 9 prompts that our pilot study suggested as answerable within 60 seconds.

Prose Tasks For controlled experimentation and to admit a contrast-based analysis, we selected prose stimuli that were analogous to the code stimuli. As prose writing fMRI studies have revealed differences in brain activation based on writing content [127; 129], we carefully developed our prose writing stimuli. First, we used a set of non-math analogies that have been shown to be useful in the teaching of mathematics [235; 236] to develop a list of terms associated with quantitative reasoning. Synonyms of these words were added to expand the search space. To generate Prose FITB prompts, we first matched the list of search words to a set of Scholastic Assessment Test (SAT) fill-in-the-blank questions and chose 17 such matches. We then replaced the blanks used in the original SAT prompt with the appropriate words from the SAT answer, selecting easier synonyms when our pilot study revealed that they might not be accessible to a wide population. We replaced the search word found in the prompt with a blank line; participants were asked to fill in that blank line. Our Prose LR prompts were generated by matching search words with a set of English as a Second Language (ESL) long response prompts and choosing 9 matching prompts.

29 out of the 30 recruited participants supplied valid inputs for the tasks. Per task, the 29 participants provided a maximum of 82 keystrokes (mean: 13) for FITB prose and 116 keystrokes

(mean: 36) for FITB code. For the LR tasks, we collected a maximum of 435 keystrokes (mean: 258) for prose and 244 keystrokes (mean: 121) for code. In general, the FITB tasks required fewer keystrokes to complete; participants had twice the time to complete the LR tasks. We observed that participants were able to write multiple complete sentences for prose tasks and to complete variable declarations, loops, and function calls in the time allotted.

4.2.3 Experimental Protocol

In this subsection, we provide details on the process that participants completed before and during their fMRI scans. During a two-hour session, we collected informed consent and safety screening information. Participants cleared to participate were given a coding quiz and psychological surveys. Participants were then shown a brief training video about the task before entering the scanner. Each machine session began with a high-resolution anatomical scan during which participants were given a text editor interface and were instructed to practice typing on the keyboard while lying inside the bore of the machine (shown in Figure 4.3). This practice typing was not recorded. Participants then completed four task blocks associated with code and prose writing: Prose FITB, Prose LR, Code FITB, and Code LR. To mitigate training and fatigue effects, we randomized both the category order and the task order. A fixation cross was presented between each question for a random 2–10s duration to provide a brief rest and settle brain activity.

4.2.4 fMRI Data Acquisition

MRI data were acquired with protocols ensuring high spatial and high temporal resolution. We summarize the details (e.g., for the purposes of replication and meta-analysis), but generally attest that the scanning measurement hardware and steps align with contemporary best practices [15; 35; 67]. All scans were conducted on a 3T General Electric MR750 scanner with a 32-channel head coil at the University of Michigan Functional MRI Laboratory, the same fMRI machine used in Chapter 3 and the technical parameters for setting up the scans are identical to the protocol in Chapter 3 (see Section 3.2.3.1).

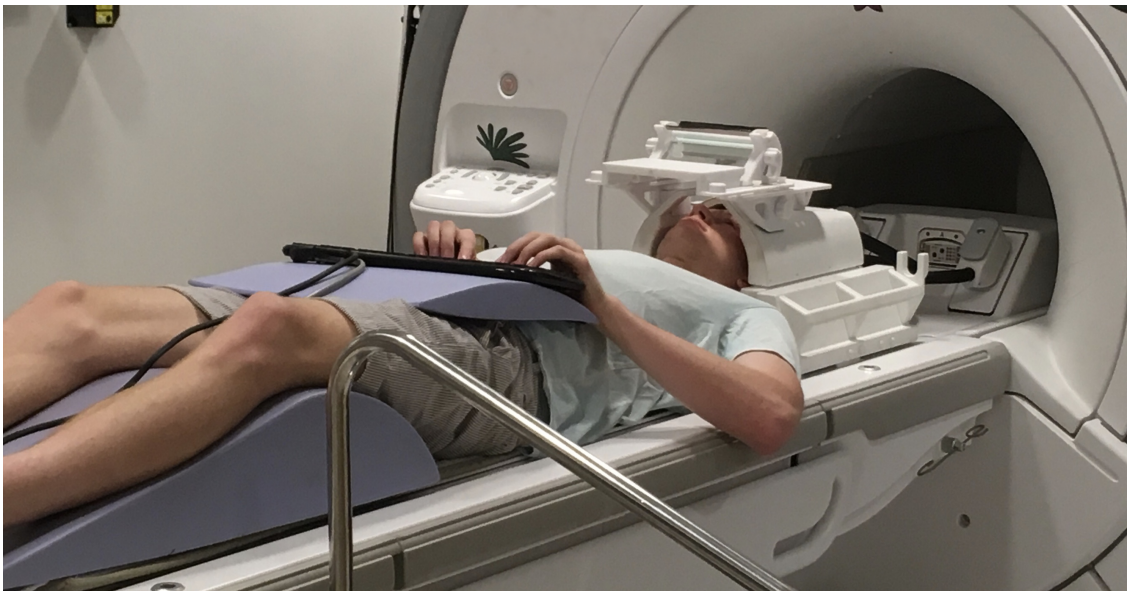


Figure 4.3: fMRI environment for typing on the bespoke keyboard. During a scan, the participant would be placed further in the bore of the machine, but the keyboard and visual interface remain as shown.

4.2.5 fMRI-Safe Keyboard and Editing

Because the fMRI machine involves an extremely powerful magnet and very strong electromagnetic fields, typical electronic devices cannot be used safely nearby. For example, a traditional USB keyboard will not function in the MRI machine because it will induce current on the USB cable, causing erratic keystroke signals or unpredictable behavior. Moreover, large metal masses within the MRI's magnetic field can cause disastrous signal noise and ruin brain images (and also pose fire and collision hazards).Previous fMRI studies of software engineering all employed special hand-held button-press devices for selecting among a small, fixed set of choices (e.g., [67; 35; 15]). These devices do not meet the requirements for code writing.

In this work, we adapted a 101-key QWERTY USB keyboard for the fMRI environment. All control logic and metal are removed from the keyboard, and moving metallic pieces are replaced with 3D-printed (plastic) equivalents. Briefly, each individual key is attached to its own shielded wire that extends 30 feet to provide adequate distance from the core of the MRI machine. The wires were fed through an RF-safe waveguide to the fMRI control room, where a custom-built device reads the state of each key and outputs a standard USB signal. Because no control logic was

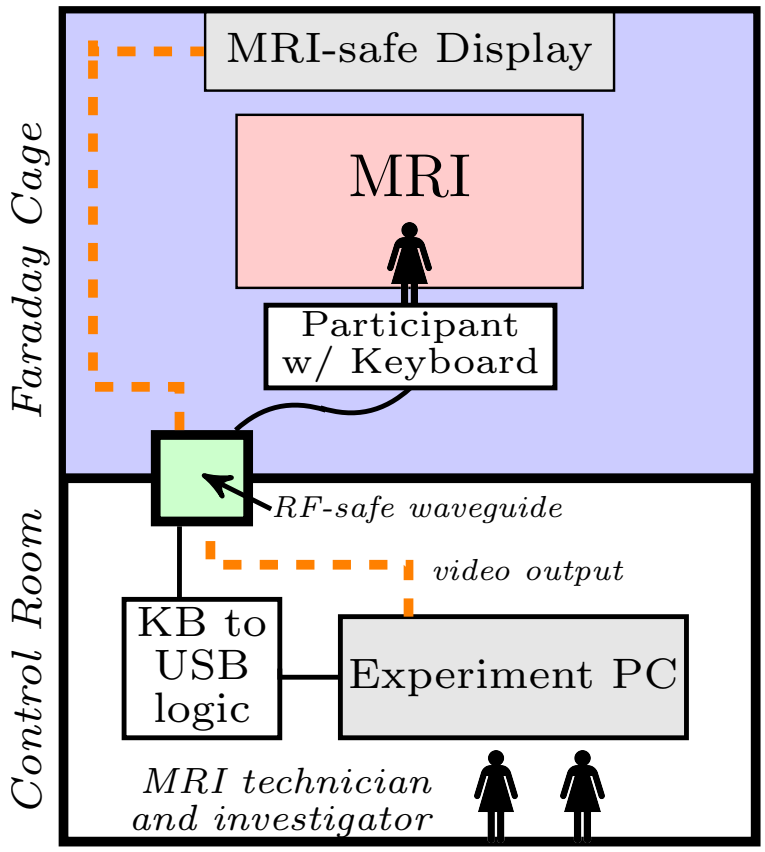


Figure 4.4: Illustration of fMRI writing setup. The participant lies in the bore of the fMRI machine, and the keyboard's cables are connected through an RF- and MRI-safe waveguide. The waveguide connects to the control room, where we attach the keyboard logic to an experiment PC displaying our editing environment. The video output of the experiment PC feeds through the waveguide to connect to an MRI-safe monitor which can be seen by the participant in the MRI bore via mirror projection.

present near the MRI machine and keystrokes were processed from the control room, we eliminated issues caused by electromagnetic interference. In addition, most fMRI studies use sequences of static pre-rendered stimuli controlled by software (e.g., E-Prime [237]) to record responses. We instead employed a more indicative dynamic editor environment, including syntax highlighting, available in our replication materials. We found this organization, illustrated in Figure 4.4, to work well, although imperfectly-printed plastic pieces caused occasionally-duplicated keystrokes for two participants.

While researchers have considered the problem for piano keyboards [238], keyboards with no screen (and thus no back-and-forth editing, *e.g.*, [130]) or significant restrictions on which keys could be pressed (*e.g.*, [239]), to the best of our knowledge, the closest related fMRI-keyboard work tends to be about a decade old (*e.g.*, [240; 130]). Existing work has primarily focused on experimental design that reduces signal noise affecting brain scan quality. Our custom-built keyboard introduces negligible signal noise on the fMRI brain scan, but also supports our additional use case for live editing during scanning. We make our engineering notes on our successful approach (and failed attempts) available as part of our replication materials.

4.3 Analysis Approach

Care must be taken when analyzing fMRI results to both mitigate noise and also to avoid false positive correlations [241]. Informally, we follow a three step process: preprocess the data to account for noise, analyze individual participants, and compare between participants. This data analysis approach follows the basic principles in Section 3.3.1 and is adjusted for the data collected from this specific experiment. Our approach follows the state-of-the-art in medical imaging (both for cognitive neuroscience in general and for software engineering in particular, *e.g.*, [15; 67; 35]). We present our results in Section 4.4; the remainder of this section summarizes our analysis for replication and comparison purposes.

Statistical analysis of fMRI data is inherently multi-level. The data first require extensive

preprocessing to remove various sources of systematic noise (*e.g.*, due to head motion or inhomogeneities in the magnetic field). An additional goal of this procedure is to align all individual participant brains with a standard anatomical template — this allows for inter-participant comparison and localization of signals to specific brain structures. Following preprocessing, each participant's data are submitted to a *first-level*, fixed effects general linear model (GLM). Here, voxel timeseries are modeled against an idealized timeseries, given the canonical hemodynamic response function and the occurrence of each *event* (i.e., stimulus) over the course of the scan. This yields a set of *beta images* that describe how sensitive each voxel is to the conditions of our experiment. Finally, the beta images for each participant are combined in a *second-level*, random effects GLM, which yields average maps of brain activity when *contrasting* one condition versus another (e.g., code vs. prose tasks). Importantly, because these statistical tests are conducted on a voxel-by-voxel basis (covering tens of thousands of voxels), we apply a *false discovery rate* (FDR) correction to protect against spuriously-significant effects across the brain.

Preprocessing — Removing Noise The preprocessing step removes noise (such as from motion during the scan). We employed a robust preprocessing pipeline using the Statistical Parametric Mapping 12 Matlab package [242]. First, the functional timeseries were slice-time corrected — this accounts for the fact that *interleaved* slice acquisition during scanning causes slight differences in the relative timing of data collection within a TR (i.e., the 800 ms interval during which whole-brain volumes are sampled). Next, we applied head motion correction and *unwarped* the data using *voxel displacement maps* derived from the fieldmap sequence (see Section 4.2.4). This step is arguably the most crucial aspect of preprocessing, as head motion is the leading cause of signal artifacts in fMRI data, further interacting with baseline distortion in the magnetic field to geometrically warp voxels. We then segmented and skull-stripped the anatomical images, which were subsequently coregistered to the functional data; both anatomical and functional scans were then spatially-normalized to the Montreal Neurological Institute (MNI152) template [210]. Finally, we constructed a *brain mask* for each participant, which ensures the exclusion of voxels outside of

brainspace during statistical analysis.

First-level Analysis — Within One Participant The first-level analysis focuses on each participant individually. We specified four GLMs for each participant (corresponding to each of the FITB and LR code and prose tasks). The onsets and durations of each trial were defined and convolved with the canonical hemodynamic response function [243] to yield a predicted timeseries of activity (i.e., how we would expect the signal in a voxel to behave if it were sensitive to our task). The data were high-pass filtered ($\sigma = 128$s) to remove low-frequency noise, and the model was fit using *robust weighted least squares* (rWLS) [244]. Since these data may be more susceptible to head motion (as a result of typing on the keyboard), we view rWLS as essential for ensuring unbiased parameter estimates: the objective function first obtains an estimate of the noise variance at each scan, and the model is subsequently re-fit after reweighting the data by a factor of 1/variance. Thus, any scans biased by head motion are given less influence in the model, allowing for homogeneous error variance and optimal parameter estimates.

Second-level Analysis — Between Participants The second-level analysis compares how different participants approached the same task. Prior to second-level GLM, the beta images for each participant were spatially smoothed using a 5 mm^3 full-width at half-maximum (FWHM) Gaussian kernel. These were submitted to an omnibus model (i.e., a factorial analysis of variance) fit using restricted maximum likelihood (ReML). To test for average differences in activity between conditions, we specified several *t*-contrasts: Code $>$ Prose, FITB Code $>$ FITB Prose, and LR Code $>$ LR Prose. The *contrast* $A > B$ refers to the comparison between task conditions A and B: voxels or features that are more sensitive to A rather than B, or that drive the modeled distinction between A and B, as introduced in 2.1.2.2. In general, fMRI cannot be used to examine a condition C directly; a subtractive controlled experiment is used instead to compute $A - B$. For example, in our experiments both the FITB and LR tasks feature reading a written prompt, but in general the neural activity associated with reading the prompt "cancels out" when the two are contrasted, and any remaining difference can be attributed to non-identical parts of the experimental

condition (i.e., writing code vs. writing prose). The ultimate result of this process is a *statistical parametric map* that displays significant contrast-related activity across the brain, quantified using *t*-statistics — the *magnitude* of the mean difference between A and B, scaled by model error. Traditionally, brain regions showing significantly more activity in A relative to B are represented with a gradient of 'hot' colors (red to yellow), while regions that are more active during B than A are represented by a gradient of 'cool' colors (blue to green). Such contrast-based analyses are standard for fMRI [67; 35; 15]. All results were FDR-corrected ($q < .05$) and thresholded for a minimum cluster extent of 20 voxels.

4.4 Results

We analyze our results with respect to four research questions:

RQ 4.1 Do self reports claim code writing is like prose writing?

RQ 4.2 Does the brain treat code writing like prose writing?

RQ 4.3 What low-level features explain code and prose writing?

RQ 4.4 What high-level features explain code and prose writing?

To guide the interpretation of our results, we consider an informal model in which long response coding (the task we studied that is most indicative of coding practice) is made up of the iterative, low-level selection of individual pieces of syntax guided by top-down control. That is, writing a small procedure (the long response task) consists of repeatedly writing the next individual word (the fill-in-the-blank task) while guided by a higher-level goal. Examining the FITB task sheds light on the lower-level basis for code writing, while examining the LR task may illuminate aspects of higher-level "creativity" at the heart of software engineering.

In our fMRI analyses, after filtering incomplete and noisy brain scans, we used data from 24 (8 female, 16 male) of the 30 participants in our experiment. When reporting patterns of neural activity we make use of the standard Brodmann anatomical classification system, which divides

the brain into 52 areas (BA 1 through BA 52) [71] based on cytoarchitectural (i.e., cellular-level) similarity. The fMRI results discussed in this section are obtained following the contrast-based analysis methodology described in Section 4.3.

4.4.1 RQ 4.1 — Self-Reporting on Code and Prose

We conducted a qualitative analysis of participants' self-reported post survey data. Of our 30 participants, 26 provided their interpretations of similarities between prose and code writing tasks in the post survey. Over a third (38.5%) of these participants reported some similarity between code writing and prose writing. Representative examples include explaining how "filling in the blank was like adding variables in code" (we investigate such similarities in Section 4.4.3) and that both tasks "use logic" (we consider mental representations and problem solving in Section 4.4.4). Another participant attributed similarity between the two tasks to having "already formed" an idea of the solution that had only to be translated to text (we consider working memory and attentional control in Section 4.4.2). As our imaging results will reveal, these subjective reports do not align with measurements of the neural correlates of code and prose writing.

Unreliable self-reporting is well-established in both computer science [16; 17; 18; 15] and psychology [51; 52], highlighting the need to augment surveys with more objective metrics.

4.4.2 RQ 4.2 — Code Writing vs. Prose Writing

We investigate whether there are *general* differences in neural activity between writing code and writing prose. We thus consider all of our code writing tasks (FITB and LR) against all of our prose writing tasks (FITB and LR). This broader Code $>$ Prose contrast, shown in Figure 4.5, revealed a widely-distributed set of brain regions showing significantly greater activity when writing code. Only significant regions are shown: the colors correspond to the t statistic, which measures the size of the difference relative to the variation in the sample data (t values closer to 0 are not significant after FDR-correction). While care must be taken when comparing such statistics across experiments, as an example baseline we note that the greatest t-value reported in the previous

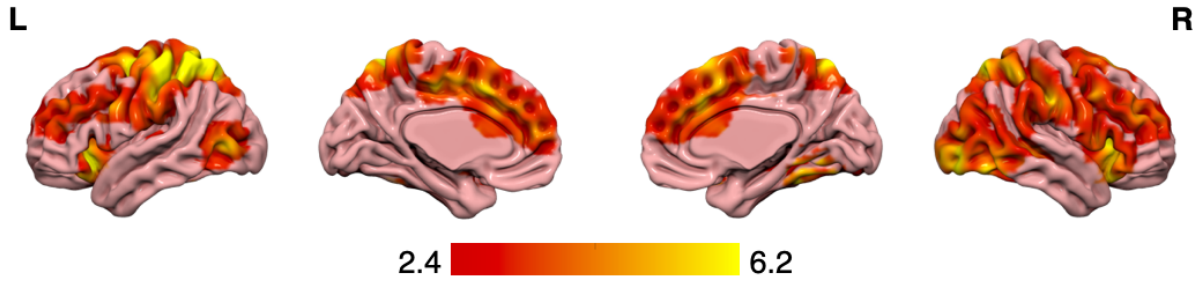


Figure 4.5: Weight map: significant clusters of brain activity for CodeWriting $>$ ProseWriting. Hotter colors indicate greater t-values (i.e., more activity during coding relative to prose).

Chapter's study of data structures and spatial ability was $2.0 \leq t \leq 4.2$. We view this $2.4 \leq t \leq 6.2$ contrast as a very strong result.

In detail, a particularly large cluster peaked near the left postcentral gyrus and superior parietal lobule (BA 5), extending forward through the primary motor cortex (BA 4) and the premotor/supplementary motor cortex (BA 6). This pattern was also observed in the right hemisphere, albeit yielding somewhat smaller differences in activity (reflected in the smaller *t*-statistics). However, the right hemisphere did demonstrate more diffuse activity through the lateral prefrontal cortex, including the superior and middle frontal gyri (BA 9–10). The right hemisphere further showed wider clusters of activity in the lateral temporo-occipital and temporoparietal cortex, spanning from the inferior and middle temporal gyri dorsally to the angular gyrus, supramarginal gyrus, and inferior parietal lobule (BA 18–19, 39–40). Finally, we observed comparable patterns of activity in bilateral anterior insula (BA 13) and across the midline of the brain, particularly the medial face of the supplementary motor area (BA 6) and the cingulum (both middle and anterior; BA 24, 32).

We find a significant ($2.4 \leq t \leq 6.2$) and widely-distributed difference in neural activity between code writing and prose writing in general. The brain does *not* treat code writing and prose writing as similar tasks.

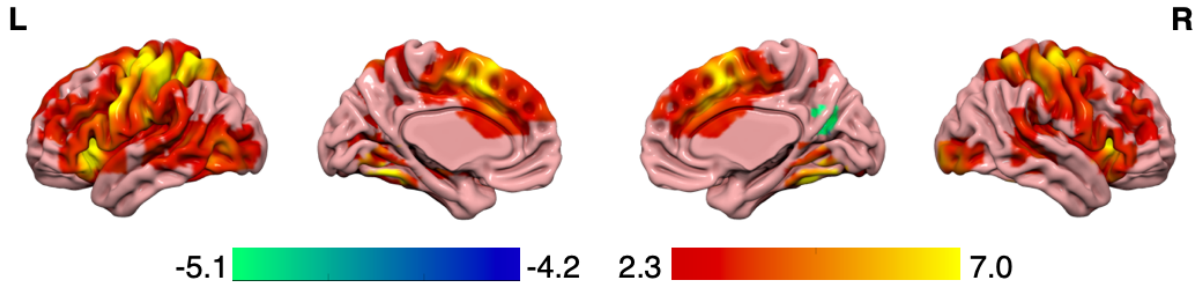


Figure 4.6: Weight map: significant clusters of brain activity for FITBCode > FITBProse. Hotter colors indicate more activity during coding relative to prose; cooler colors indicate the reverse.

4.4.3 RQ 4.3 — Code and Prose Foundations

Having established that the brain treats code writing and prose writing differently, we focus attention on our lower-level tasks to explain that difference. We thus consider the contrast FITB Code > FITB Prose, shown in Figure 4.6. While there was considerable overlap between this contrast and the general Code > Prose analysis (informally, we expect some similarity between writing one word and writing a full sentence), we find that focusing on FITB Code > FITB Prose reveals even stronger ($-5.1 \leq t \leq -4.2$ and $2.3 \leq t \leq 7.0$; conservatively thresholded for multiple comparisons) differences in activity across a number of regions. For example, we observed strong bilateral activity across the entirety of both precentral and postcentral gyri (i.e., the primary motor and somatosensory cortices, respectively; BA 1–4). While these areas are essential for somatomotor function, they are not *cognitive* — that is, activity in these regions does not directly involve 'thought' or 'planning'. These aspects of motor behavior are generally supported by the dorsal premotor cortex and (pre-)supplementary motor area (BA 6, 8), which show significantly greater bilateral activity when performing FITB Code vs. Prose trials. *This suggests that the production of even a single element of code may require more careful, top-down control to effectively plan and produce a context-relevant answer.* This is further supported by significant differences in activity along the frontal eye fields, including the prefrontal eye fields and supplementary eye fields (BA 8–9): these regions are known to help guide the eyes toward relevant stimulus features to generate

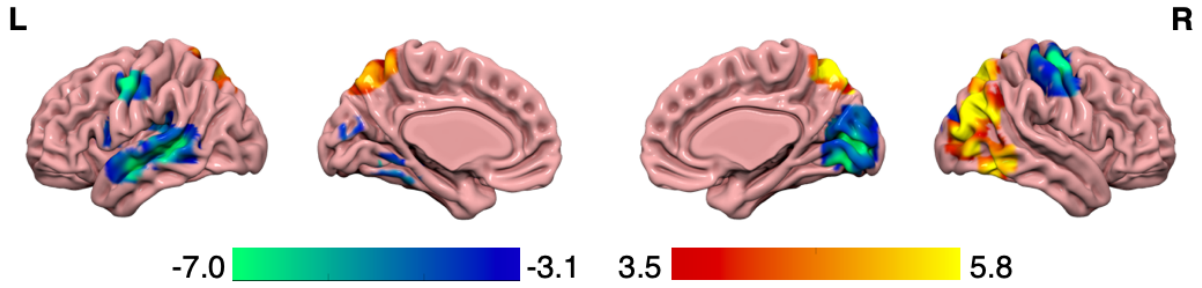


Figure 4.7: Weight map: significant clusters of brain activity for LRCode $>$ LRProse. Hotter colors indicate more activity during coding relative to prose; cooler colors indicate the reverse. This represents a strong and exciting result: a significant lateralized difference between prose writing (canonical left hemisphere language areas) and code writing (right hemisphere attention, memory, planning and spatial ability areas).

an appropriate motor plan [245; 246].

We additionally observed significant increases in activity within other regions comprising the so-called 'dorsal attention network' (of which the frontal eye fields are a part). This includes the superior parietal lobule and intraparietal sulcus (BA 7) — structures critical for guiding and maintaining attention in a top-down fashion [247; 248]. Although not part of the dorsal attention network, the bilateral activity found in the anterior insula (BA 13) further supports the notion that FITB Code likely requires more careful monitoring of the relevant information needed to provide the appropriate response.

Finally, we note significant differences in activity along posterior temporal/occipital-temporal regions. In general, these appear left dominant, although bilateral activations emerged in the posterior superior temporal gyrus and superior temporal sulcus (BA 21–22). Interestingly, we also observed bilateral activity in the ventral temporal cortex, including the fusiform gyrus (BA 20, 37). While the fusiform gyrus is perhaps best known for its role in face perception, it (along with other areas of the ventral temporal cortex) is also heavily involved in stimulus categorization, particularly for stimuli with which one has developed expertise. This poses the possibility that code — despite being a collection of numbers, letters, and words — is nevertheless treated as a categorically distinct visual stimulus compared to English prose.

At a low level, writing code requires significantly ($-5.1 \leq t \leq -4.2$ and $2.3 \leq t \leq 7.0$) more activity in parts of the brain associated with careful top-down control, planning, and categorization than does writing prose.

4.4.4 RQ 4.4 — High-Level Coding vs. Prose Writing

Finally, but perhaps most excitingly, we analyze long response code and prose writing tasks. Long response tasks (*i.e.*, writing an entire method) are the most indicative of critical aspects of real-world software engineering. If we consider long response coding to include both the iterative production of single code elements as well as top-down attentional cover of the overarching process, then any difference between this analysis and RQ3 reveals the neurological correlates of that high-level "creativity" in coding.

Figure 4.7 shows the LR Code $>$ LR Prose trials contrast. This analysis remains strongly significant ($-7.0 \leq t \leq -3.1$ and $3.5 \leq t \leq 5.8$; conservatively thresholded for multiple comparisons) and more precisely pinpoints particular regions. Note how the regions associated with high t-values (hotter colors, more active for code than prose) are largely localized to the right side of the brain. Dually, note how the left hemisphere largely features regions with very low t-values (cooler colors, more active for prose than code). In cognitive neuroscience, such a left vs. right distinction is called *lateralization*. These contrast-based results provide powerful evidence that the *production* of code vs. prose relies on highly distinct cognitive substrates.

Prose production was strongly associated with left temporal regions classically associated with natural language (which is almost entirely left-lateralized in right-handed individuals). Namely, we saw increased recruitment of the middle temporal gyrus (MTG) and superior temporal gyrus (STG) (BA 21–22). The left MTG has previously been shown to activate when accessing semantic aspects of language and is thought to support a lexicon of words [249; 250]. The STG extends into Wernicke's area, which is notably the primary center of language comprehension [251]. Although it generally appears most active during comprehension of *spoken* language, the act of writing

often involves a sort of internal narration that may similarly recruit these regions. This is further supported by increased activation of the calcarine (visual) cortex, particularly the lingual gyrus along the right medial wall (BA 17–18). The lingual gyrus, while not playing a role in higher-order language processes *per se*, is often associated with the recognition of letters and words, perhaps contributing to their semantic understanding [252]. We also observed a small cluster of activity in the inferior frontal gyrus (BA 44) — part of Broca's area, which underlies the production of language (although, again, is more commonly linked to speech) [253].

Code production, by contrast, was largely right-lateralized. The exception to this observation was a bilateral activation of the superior parietal lobule, extending dorsolaterally into the precuneus along the midline (BA 7). The superior parietal lobule (see Section 4.4.3) is involved in top-down control processes related to attention and memory; the precuneus is associated with processes such as mental imagery [254]. Similarly, we observed right temporal and temporoparietal activations along a number of regions supporting visual association (tying visual information together) and other forms of mental imagery, including spatial cognition (BA 19, 39). The angular gyrus, in particular, may support various aspects of spatial and mathematical reasoning, including the manipulation of mental representations and other aspects of problem-solving [255]. Importantly, it is thought to act as a bridge between perception, recognition, and action, suggesting that code synthesis may require a more complex interplay of understanding a problem and formulating a comprehensive plan to solve it [256]. This swath of activity extended ventrally into regions of the inferior occipital-temporal cortex, which partially overlap with clusters identified by Huang *et al.* as being more active during difficult data structure manipulations (relative to difficult mental rotation tasks) [15]. Together, these findings suggest that code production is perhaps more 'spatial' in nature, requiring the formulation of a mental map that guides problem-solving.

Very informally, finding activity in the (expected, standard) language areas for prose writing gives us high confidence that we designed and carried out our controlled experiment correctly in general. However, that high confidence makes the observation that long-form code writing does not heavily recruit these areas (instead using parts of the brain associated with planning and

spatial ability) all the more startling. Part of the motivation for Siegmund *et al.*'s pioneering first use of fMRI in software engineering [35] was to provide direct evidence, one way or another, regarding claims such as Dijkstra's that "exceptionally good mastery of one's native tongue is the most vital asset of a competent programmer" [257]. While that may be true for code reading (e.g., comprehension [35] and reviewing [67]), our results suggest that it is *not* true for code writing at a neural level.

> High-level long response coding is significantly different ($-7.0 \leq t \leq -3.1$ and $3.5 \leq t \leq 5.8$) from prose writing. Prose writing involves areas of the brain canonically associated with language. Coding involves a different set of right-lateralized regions associated with attention, memory, planning, and spatial ability. This provides the first evidence of significant neural differences between *prose writing* (which is neurally similar to natural language) and *code writing* (which, we find, is *not*).

4.4.5 Summary of Results

At a high level, an analysis of all code writing tasks against all prose writing tasks showed that the two operate via distinct neural mechanisms. We analyzed these differences at a more granular level by considering imaging data from tasks of the same type (i.e., FITB, LR). The FITB Code $>$ FITB Prose contrast established the low-level cognitive features distinct to code writing: brain regions associated with top-down control, planning, and categorization. Subsequent analyses of LR tasks revealed a clear lateralized distinction between code writing and prose writing. Largely, we found that code writing involves right hemisphere brain regions involved in spatial ability and planning while prose writing involves the canonical left hemisphere regions associated with language production. In addition to supporting previous medical imaging studies of prose writing and software engineering tasks, these findings introduce a new and alternative relationship between code and prose in which reading and writing are *not* similar (cf. [257; 35; 67]).

4.5 Threats to Validity

Our choice of writing tasks presents a first potential threat to the validity of our experiment. While various forms of code writing exist in software engineering contexts (e.g., testing, debugging), we restricted our task set to *prompted* code writing tasks. We mitigate the threat that this limited benchmark poses via our robust experimental design, whereby participants complete different types of writing tasks (i.e., FITB, LR). We further address this concern by including a variety of fundamental programming concepts (e.g., both control- and data-flow operations) in our selected coding tasks indicative of many real-world coding tasks. Nevertheless, our results may not generalize to all in-the-wild programming; we leave a more thorough investigation to future research.

Secondly, the design of our tasks may have impacted our ability to measure brain activity strictly associated with code and prose writing. For example, our stimuli included written instructions that participants read before typing their responses. This construction introduces the possibility that we measure brain activity beyond strictly writing responses. We designed our contrast-based experiment to mitigate this threat. As fMRI analyses are subtractive (described above in Section 4.4), the effects of reading the prompt cancel out, leaving only the differences between prose writing and code writing. However, we note that differences exist in the prompt text contained in FITB stimuli (i.e., Prose FITB and Code FITB tasks require the participant to read prose and code, respectively, see Figures 4.2a and 4.2c). Overall, we maintain that FITB tasks measure the process of low-level selection of individual code elements, a distinct activity to pure comprehension.

Lastly, our results may be limited by our participant cohort. For this experiment, we recruited undergraduate and graduate students with an average of 5.2 semesters of programming experience. Thus, our results may not extend to programmers with different backgrounds or expertise. Indeed, previous fMRI studies have investigated the role of expertise and demographics in detail (e.g., [67; 258]). We claim no significant findings regarding individual differences and report results for our participants as a whole.

4.6 Chapter Summary

Over the decades, researchers from Dijkstra and Pausch to Pea and Kurland, among many others, have made observational investigations into, theories of, and calls to arms regarding the psychological aspects of programming. Understanding cognition has helped improve prose reading, prose writing, and code reading — but code writing has lacked neurologically-grounded, indicative research. Indeed, since the first fMRI study of software engineering in 2014, there have been fewer than 20 fMRI experiments reported at major SE conferences as of 2021 [35; 67; 173; 174; 38; 37; 36; 175; 176; 176; 15].

We present the first fMRI study of code writing. We employ a controlled, contrast-based experiment in which code writing, prose writing, fill-in-the-blank and long response tasks are presented to participants, who make use of a special fMRI-safe keyboard to type their responses in a realistic live editing setting.

We report three primary results. First, there is a significant and widely-distributed difference in neural activity between code writing and prose writing in general: the brain does *not* treat code writing and prose writing as similar tasks. Second, at a low level (i.e., producing a single word or code element), writing code requires significantly more activity in brain areas associated with careful, top-down control, planning and categorization: despite superficial similarity, code appears to be a categorically distinct visual stimulus compared to prose. Third, and most excitingly, high-level long response coding — the studied task perhaps most indicative of real-world programming — is significantly different from prose writing. While prose writing involves left-brain regions canonically associated with language, we find a sharp lateralized distinction: code writing does *not* significantly recruit those regions compared to prose writing, instead showing activation in right-brain areas associated with attention, memory, planning and spatial ability.

This unexpected result — that the production of code and prose rely on highly distinct cognitive substrates — though quite preliminary, paves the way for future investigations analogous to those based on medical imaging for prose writing. In addition to developing a foundational understanding of code-writing, this empirical distinction may be leveraged to develop tools and pedagogies

(e.g., transfer training), subsequently affecting large scale workforce retraining and educational reform. Moreover, neurological evidence that code and prose writing are not as intertwined as conventionally thought may encourage more diverse participation in computer science.

In this chapter, we investigated the cognitive processes in a higher level software engineering task, code writing, using fMRI and a bespoke keyboard. In the next chapter, we study cognitive processes and biases across different demographic groups in a critical software development task: code review. We also present the usage of eye tracking in the next chapter.

CHAPTER 5

Bias in Code Review Across Groups of Users

In previous chapters, we investigated the cognitive processes of data structure manipulations and code writing using medical imaging techniques such as fMRI and fNIRS. In this chapter, we introduce the study of cognitive processes and biases across different demographic groups in a critical software development task: code review. We also introduce the setup and results from a combination of fMRI and eye tracking measurements.

Code review, first introduced in Chapter 2.3.4, is a common and critical practice in modern software engineering for improving the quality of code and reducing the defect rate [57; 58; 55; 56]. Generally, a code review consists of one developer examining and providing feedback for a proposed code change written by another developer, ultimately deciding whether the change should be accepted. In modern distributed version control, code review often centers around the *Pull Request* (or *merge request*) mechanism for requesting that a proposed change be reviewed. The importance of code review has been emphasized both in software companies (e.g., Microsoft [259], Google [260; 261], Facebook [262; 263]) and open source projects [264; 265]. While code review is widely used in quality assurance, developers that conduct these reviews are vulnerable to biases [40; 41]. In this chapter, we investigate objective sources and characterizations of biases during code review. Figure 5.1 shows a high-level view of our study: does the authorship of a Pull Request influence reviewer behavior, and do men and women evaluate Pull Requests differently? Such an understanding may help reduce bias to improve developer productivity.

While there are many potential sources of bias in code review (including perceived exper-

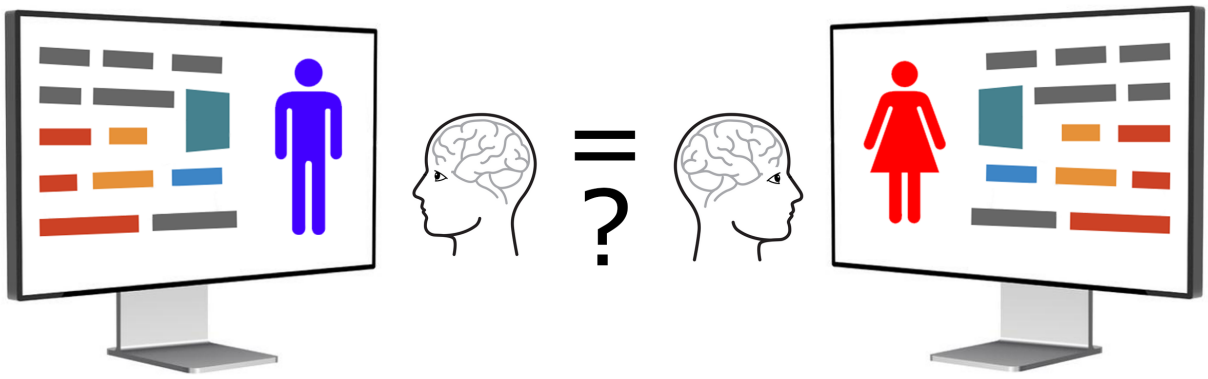


Figure 5.1: Illustration of the investigation on biases in code review: we investigate the relationship between code review activities, participants and biases. Experimental controls systematically vary the labeled author (man vs. woman vs. machine) while controlling for quality.

tise [266], perceived country of origin [267], and reviewer fatigue [268]), of particular interest are biases associated with the perceived gender of the author. These are relevant from a moral perspective (e.g., broadening participation in computing [269]), from a process efficiency perspective (e.g., arriving at the correct code review judgment [270]), and even from a market perception perspective (*e.g.*, recent scandals involving gender-fairness in hiring and development processes [271; 272]).

Previous studies have shed light on the effects of gender bias in software development by analyzing behavioral data. For example, large-scale analyses of GitHub Pull Request data found that women's acceptance rate is higher than men's when their gender is not identifiable, but the trend reverses when women show their gender in their profiles [40]. Similarly, another study using behavioral data on GitHub found that women concentrate their efforts on fewer projects and exhibit a narrower band of accepted behavior [39]. Furthermore, research has shown that developers may not even recognize the potential effects of biases of code authors when performing code reviews [41; 40]. Such biases may not only decrease the quality of code reviews, but also the productivity of software development, especially in fields like software engineering that are dominated by men [273; 274; 275] despite (gender) diversity significantly positively influencing productivity [267; 165; 59; 60].

Moreover, not all code changes are generated by humans. From 2010 to 2020, there has been a flurry of research into Automated Program Repair (APR) tools in both academia and indus-

try [276; 277]. As of 2020, APR tools have seen increased adoption among larger (*e.g.*, Facebook's SapFix [278]) and smaller (*e.g.*, Janus Manager [279]) companies. However, many developers express reluctance about incorporating machine-generated patches into their code bases [280] and expert programmers are less accepting of patches generated by APR tools [281]. In such situations, human biases may interfere with the potential business benefit associated with the careful deployment of such automation [278; 279; 277; 282].

Unfortunately, research studying how developers perceive and evaluate patches as a function of their *provenance* (i.e., source or author) has been limited. Although the software engineering community has realized the importance of overcoming the negative effects of bias [267; 165], we still lack a fundamental understanding of how bias actually affects the cognitive processes in code review. This lack of objective basis in understanding bias hinders the development and assessment of effective strategies to mitigate productivity and quality losses from biases in code review.

In the psychology literature, researchers have explored the effects of bias in myriad daily life scenarios. For example, behavioral studies have revealed biases in gender and race in fields such as the labor market [283], self-evaluations of performance [284], publication quality perceptions and collaboration interest [172], online product reviews [285] and peer reviews [171; 286; 287]. Furthermore, psychologists have also adapted medical imaging techniques to investigate the cognitive processes associated with bias in different activities. In controlled experiments using medical imaging techniques, psychologists have found several specific brain regions that are associated with bias in humans' cognitive processes [288; 289; 290; 291; 292; 293; 294; 295]. These psychology studies provide a model for the investigation of the behavioral and neurological effects of biases in software development tasks.

5.1 Overview of Experimental Design, Results, and Contributions

Our experiment involves measuring humans as they conduct code review. In particular, we make use of a controlled experimental structure in which the same code change is shown to some participants with one label (e.g., written by a man) but is shown to other participants with a different label (e.g., written by a woman or machine). Beyond measuring behavioral outcomes (e.g., whether or not the change is accepted, how long the review takes, etc.), we also use fMRI, which enables both the analysis of neural bases underlying code review activities and also the inference of biases (if they exist).

However, fMRI does not provide significant evidence about participants' visual interaction with the code itself. We build on previous work and address this problem by capturing participants' attention patterns and interactions via *eye-tracking*, which has been used to understand developers' visual behavior in code reading [296; 297; 298] as well as the impact of perceived gender identity in code review [41]. Using eye-tracking in combination with fMRI allows assessing both neural activity and higher-level mental and visual load in human subjects as they complete cognitive tasks.

We desire an understanding of code review that (1) explicitly incorporates gender bias, (2) is based on multiple types of rigorous physiological evidence, and (3) uses controlled experimentation to provide support and guidance for actionable bias mitigations. Previous studies have considered these goals pairwise, but not all simultaneously. For example, there have been behavioral studies in both computer science and psychology on biases (e.g., [40; 39; 283]), medical imaging studies of biases in psychology (e.g., [288; 291]), eye-tracking studies of biases [41], and eye-tracking [105; 104] and medical imaging studies [67; 35; 15] of other factors in computer science. However, to the best of our knowledge, we present the first experimentally-controlled study investigating biases in computing activities by measuring multiple neurophysiological modalities.

Contributions. We present the results of a human study involving 37 participants, 60 GitHub Pull Requests, three provenance labels (man, woman, and machine), fMRI-based medical imaging,

and eye-tracking. Men and women participants conduct code reviews differently:

- Behaviorally, the gender identity of the reviewer has a statistically significant effect on response time ($p < 0.0001$).
- Using medical imaging, we can classify whether neurological data corresponds to a man or woman reviewer significantly better than chance ($p = 0.016$).
- Using eye-tracking, we find that men and women have different attention distributions when reviewing ($p = 0.005$).

In addition, we find universal biases in how all participants treat code reviews as a function of the apparent *author*:

- Participants spend less time evaluating the pull requests of women ($t = -2.759$).
- Participants are more likely to accept the pull requests of women and less likely to accept those of machines ($p < 0.05$).
- Even when quality is controlled, participants acknowledge a bias against machines (˜$3\times$), but do not acknowledge a gender bias (even as evaluation and acceptance differ).

We also make our dataset available for analysis and replication.

5.2 Experimental Setup And Methods

We present a human study of 37 participants. In our experiment, every participant underwent an fMRI scan and eye-tracking simultaneously while completing code review tasks. The eye tracker is integrated into the fMRI machine and two sets of fMRI-safe buttons were positioned in each of the participant's hands to record inputs. In this section, we discuss (1) the recruitment of our participants, (2) the preparation of our code review stimuli, (3) the experimental protocol, and (4) our fMRI and eye-tracking data collection methodology.

Table 5.1: Demographic data of the eligible participants in the study of biases in code review.

Demographic	Number of Participants		
	Total	Version I	Version II
Men	21	11	10
Women	16	7	9
Undergraduate	26	11	15
Graduate	11	7	4

All of our de-identified data are available at our project website[1].

5.2.1 Participant Demographics and Recruitment

Table 5.1 summarizes demographic information for our participant cohort. We recruited 37 undergraduate and graduate computer science students at the University of Michigan; the study was IRB approved. We required participants to be right-handed with normal or corrected-to-normal vision, and to pass a safety screening for fMRI. In addition, we required participants to have completed data structures and algorithms undergraduate courses. Participants were offered $75 cash incentives and scan data supporting the creation of 3D models of their brains upon completion.

5.2.2 Materials and Design

Participants underwent an fMRI scan and eye-tracking during which they completed a sequence of code review tasks. More specifically, a single code review task consisted of evaluating an individual Pull Request and deciding whether to *accept* or *reject* the proposed changes. Participants were shown a sequence of Pull Requests adjusted to fit the fMRI's built-in monitor. The technical contents of the Pull Requests (e.g., the code change, context, and commit message) were taken from historical GitHub data; the identifying information (e.g., purported names and faces of developers) was experimentally controlled. We designed the code review stimuli following the best practices in previous fMRI research in software engineering [41; 67] as well as previous work in Chapter 4.

[1] https://web.eecs.umich.edu/ weimerw/fmri.html

Each code review stimulus consisted of a loading image that displayed an author profile followed by the corresponding Pull Request. Each loading image was presented for 5 seconds and each Pull Request page was presented for 25 seconds. A red-cross fixation image randomly ranging from 2-10 seconds was presented between code review stimuli.

Pull Requests: In our study, we included 60 real-world Pull Requests in total from open source C/C++ projects on GitHub. These 60 Pull Requests consisted of (1) 20 code review stimuli adopted from a previous fMRI study conducted by Floyd *et al.* [67] and (2) 40 Pull Requests obtained from the top 60 starred C/C++ projects on GitHub in February 2019. For each of the 60 GitHub projects, we requested the 60 most recently committed Pull Requests on February 3, 2019, retaining that contained (1) no more than two files with changes, (2) fewer than 10 lines of changes (to fit the fMRI monitor), and (3) at least one C/C++ file being changed. Finally, we randomly selected 40 Pull Requests from 18 different GitHub projects that meet the filtering requirements. The 60 Pull Requests have an average of 8.7 lines of code ($\delta = 1.8$) and an average of 2.7 lines of changes ($\delta = 1.5$).

Author Profile Pictures: We used human photos from the Chicago Face Database [299], which are controlled for race, age, attractiveness, and emotional facial expressions. To avoid bias from other variables of human faces, we randomly selected 20 pictures each for white women and men between 22 and 55 years old with neutral emotional facial expressions and average attractiveness ($\bar{x}_{attractiveness} \pm \sigma$). Then we conducted equivalence hypothesis tests [300] of age and attractiveness between the men and women picture sets. Both tests were significant ($p < 0.01$, using the $20\% \times \bar{x}$ bound) which indicated there was no significant difference between the women's and men's pictures with respect to age and attractiveness.

Code Review Stimuli Construction: We designed two versions of code review stimuli in this study. Each version contained 60 code review tasks which were constructed with the 60 selected Pull Requests, 40 human photos and a computer avatar (examples shown in Figure 5.5). In Version I, we randomly paired the Pull Requests and author profile pictures so that the final set of code review tasks contained 20 Pull Requests labeled as being written by women, 20 Pull Requests

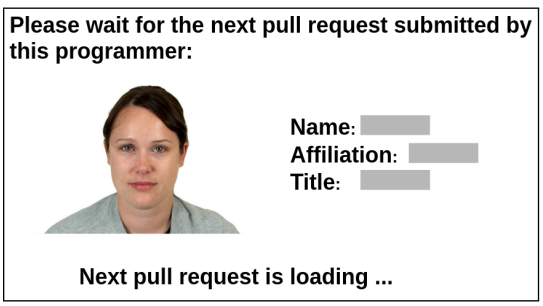
written by men, and 20 Pull Requests generated by machines (automated repair tools). Then in Version II, we relabeled all the Pull Requests, assuring that each received a different author label than in Version I while preserving a 20/20/20 split. For example, a Pull Request paired with a woman's picture in Version I would be paired with a man's picture or the computer avatar in Version II.

This two-Version approach supports our experimental control. No single participant is shown the same patch twice. However, across the entire experiment, each patch P will be constructed with two different author labels and shown once to all participants. For example, Participant A will review patch P with a man author, while Participant B will review P with a woman author. Since the technical content of patch P remains constant and only the label changes, given enough samples, differences in responses to patch P can be attributed to differences in the labels.

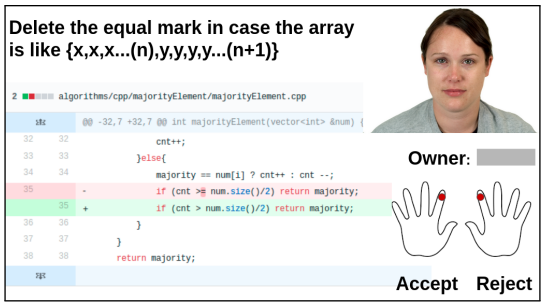
Each code review task started with a 5-second loading image that briefly introduced the purported author (shown in Figure 5.2a). The loading image also showed a grayed-out area indicating that the author's name, affiliation, and title were omitted for privacy protection. Participants were then presented with the Pull Request contents for 25 seconds (similarly, the author's name was grayed out). An example of a code review stimulus is shown in Figure 5.2b. On the bottom right corner of each code review stimulus, we displayed an indicator image to remind participants of which finger buttons to press to accept or reject the current Pull Request. This stimulus structure is broadly similar to that used by Ford *et al.* [41].

5.2.3 Experimental Protocol

We recruited participants via email lists and in-class invitations. Candidate participants were required to complete an fMRI safety screening (e.g., age between 18 and 65, right-handed, correctable vision, etc.). Each participant was also required to complete a pre-scan survey to assess minimum coding competence. We split participants into two approximately equally-sized groups of men and women. Participants in each group received either the Version I or Version II stimuli. Table 5.1 summarizes demographic information for each group. Participants gave informed



(a) Example loading image.



(b) Example code review stimulus.

Figure 5.2: Examples stimuli of code review, including a loading image (top) shown for 5 seconds before a Pull Request with author profile picture (bottom).

consent and could withdraw from the study at any time. Scans required 60–70 minutes.

Pre-scan Surveys: After participants elected to participate in the study, we first collected basic demographic data (sex, gender, age, cumulative GPA, and years of experience). We also administered a short programming quiz to assess basic C/C++ programming skills. Participants could only proceed with the study if they answered all the questions in the programming quiz correctly.

Training: We showed each participant a training video explaining the study design and purpose. Because many view gender bias as a moral or social issue, we expect that telling participants that gender bias was being studied would influence their behavior [301]. Thus, by design, we (deceptively) described this study only as *understanding code reviews using fMRI* and involving only code reviews from real-world software companies. We claimed the researchers had merely adjusted the stimuli presentation to fit the fMRI environment. We told the participants that the goal of this study was to understand how programmers think when deciding to accept or reject a Pull Request. We *explicitly* elided any mention of author gender or provenance as a basis for evaluating Pull Requests. Per IRB regulations, this *deception* required a formal *debriefing* session upon completion of the experiment to explain the true motivation of the study.

fMRI Scan: After consenting, participants underwent an fMRI scan, during which they completed four blocks of code review tasks. Additionally, we used an eye-tracking camera to record

gaze data. Each block contained 15 randomly-ordered code review tasks and 2 dummy stimuli for eye calibration that were presented at the beginning and middle of a block. For each code review task, participants were asked to review the Pull Request as a real-world software developer and use the fMRI-safe buttons positioned in their hands to provide a binary decision: accept or reject that Pull Request.

Post-scan Surveys: After the fMRI scan, participants were asked to take an Implicit Association Test (IAT) [302]. Such assessments are widely used in both psychology and engineering for investigating implicit, relative associations between liberal arts and women and between science and men [41; 303]. Then, participants finished a paper-based post-survey regarding the experiment (see section 5.4.4).

Debriefing: After completing the experiment, we formally debriefed participants about the true motivation of the study. In particular, we disclosed to each participant the nature of the experiment was to evaluate gender-based biases, and that in fact the author identity information associated with each Pull Request did not correspond to actual authors. Additionally, we explained that knowing the nature of the experiment a priori might introduce social desirability bias [301].

We conducted a correlation analysis between psychology measures from pre-scan surveys (i.e., SES data), IAT results from post-scan surveys, behavioral data, eye data, and brain activity. While no simple correlations survived a significance test ($p < 0.05$), we report other significant findings in Section 5.4.

5.2.4 Data Collection

fMRI acquisition: MRI data were acquired with protocols ensuring high spatial and high temporal resolution. We summarize the details (e.g., for the purposes of replication and meta-analysis), but generally attest that the scanning measurement hardware and steps align with contemporary best practices [15; 35; 67]. All scans were conducted on a 3T General Electric MR750 scanner with a 32-channel head coil at the Functional MRI Laboratory at the University of Michigan (the same facility and parameters as used in Chapter 3 and 4).

Eye-tracking Acquisition: We used an MRI-compatible Avotec RE-5701 eye tracker to monitor and track participants' eye movements while undergoing an fMRI scan. Using a slide projector and a galvanometer-driven mirror, stimuli were back-projected onto a screen on top of the head-coil. The mirror reflected the picture of a computer screen with a resolution of 1920x1080 with fonts sized to approximately 36 pixels in height. Participants viewed the stimuli via a mirror while supine and a second mirror reflected images of the eyes to the eye tracker, installed at the head end of the scanner.

5.3 Modeling Approach

In this section we describe the mathematical modeling applied to our measurements. Key considerations include accounting for noisy physiological data, correcting for multiple comparisons (i.e., avoiding spurious conclusions resulting from repeated analysis attempts), and statistical significance.

5.3.1 fMRI Anaylsis

This preprocessing and first-level analysis below in this fMRI analysis approach follows the basic principles in Section 3.3.1 and is adjusted for the data collected from this specific experiment. In this Chapter, we adapt the *Gaussian Process Classification* to investigate the relationship on general brain patterns between men and women participants.

Preprocessing: Functional MRI data require careful *preprocessing* prior to statistical analysis: these procedures correct systematic sources of noise in the signal (e.g., due to head motion) and spatially align brains to a standardized anatomical space. Here, we implemented a robust preprocessing pipeline using the Statistical Parametric Mapping 12 (SPM12) software in Matlab. First, we used the RETROICOR technique to remove signal confounds associated with cardiac and respiratory noise. We then slice-time corrected the blood oxygen-level dependent (BOLD) timeseries to account for minor differences in the relative timing of signal acquisition within a TR

(i.e., the 800 ms window during which the whole brain is sampled). Images were then realigned to correct for head motion during the scan, and geometric deformations (due to motion and magnetic field inhomogeneity) were unwarped using data from the fieldmap sequence. Finally, we skull-stripped the high-resolution anatomical image, coregistered it with the functional data, and spatially-normalized all images to the standard MNI152 template.

First-level analysis: Task-related changes in BOLD activity were assessed on a *within-subject* basis using the general linear model (GLM). For each of the four scanning runs, we specified regressors corresponding to the author 'prime' (i.e., the 5s loading screen preceding each Pull Request) and the code review block (Pull Requests with author labels), separated by author identity (e.g., 'Man Prime' and 'Man PR'). This yielded six event types per scanning run, with review block durations defined by the participant's response time. The design matrices were convolved with the canonical hemodynamic response function (HRF) and data were high-pass filtered ($\sigma = 128$ s) to remove low-frequency noise. Model parameters were estimated using restricted maximum likelihood (ReML) with *robust weighted least squares* (rWLS) [244]: this technique ensures maximally-unbiased parameter estimation by first estimating the residual noise variance associated with each image and subsequently re-weighting scans by a factor of 1/variance. Thus, noisy images (e.g., those contaminated with motion artifact) are given less influence in the model.

Following model estimation, it is necessary to compute *contrasts* in brain activity: task-related changes in the BOLD signal can only be understood *relative* to other conditions in the experiment (see Section 2.1.2.2). Here we generated contrasts for all pairwise comparisons between author prime and code review conditions. For example, $WomanPrime > ManPrime$ and $WomanPR > ManPR$. In subsequent analyses, however, we focus on the $WomanPR > ManPR$ contrast because it represents a direct comparison in brain activity related to author gender (note that the reverse $ManPR > WomanPR$ is symmetric about zero, and therefore it would only flip the sign of the estimated parameters in our machine learning model — *not* change the fit or the results). These contrast maps for each participant were smoothed with a 5 mm^3 full-width at half maximum (FWHM) Gaussian kernel prior to group-level analysis.

Gaussian Process Classification: To test the hypothesis that men and women participants differentially process code written by women versus men, we implemented a multivariate pattern analysis using Gaussian Process Classification (GPC). Machine learning techniques such as GPC can be more powerful than conventional *mass-univariate* analyses because they harness the multivariate nature of fMRI data: rather than estimating voxel-by-voxel models of differences in brain activity (requiring conservative corrections for multiple comparisons), GPC considers *whole-brain patterns* of activity that may distinguish between groups or stimulus categories. For this analysis, we used the Gaussian Processes for Machine Learning (GPML) software v3.5 in Matlab.

The details of our approach follow Floyd *et al.*'s previous use of GPC in a software engineering context [67]. In short, the extremely high-dimensionality of fMRI images (tens of thousands of voxels) requires that data be compressed into a *feature space*. We used a simple linear kernel, whose elements indicated the degree of similarity (the dot product) between all pairs of images. A key advantage to the linear kernel — as opposed to nonlinear methods, such as the radial basis function — is the ability to project model hyperparameters back into the original data space, yielding a spatial representation of the decision function (i.e., brain regions where greater activity pushes the classifier towards predicting 'man' or 'woman'). Classification is ultimately a two-step procedure: the model is first trained to identify patterns that distinguish between men and women participants, and performance is then tested using a new image without a class label. We therefore implemented a leave-one-out cross validation scheme, where participants were iteratively removed from the training data, models were fit, and a predicted class was obtained for the left-out participant. This yields a percent classification accuracy for each group and the average *balanced accuracy* (*BAC*) of the classifier on the whole. To determine whether performance was significantly greater than chance, we ran 1,000 iterations of nonparametric permutation testing: in this procedure, class labels were randomly permuted, the entire cross-validation scheme was performed, and classification accuracies were recorded to build empirical null distributions for classifier performance. Performance is considered significant if the true model outperformed the random models more than 95% of the time.

5.3.2 Eye-Tracking Analysis

Preprocessing: Preprocessing eye-tracking data includes removing outliers and fixing offsets. An *offset* is the difference in the location of a sampled gaze point and its true coordinates, offsets grow when the participant's head falls outside the range of camera or as a result of calibration deterioration over time. We use Ogama[2] to manually identify horizontal and vertical offsets by replaying the eye gaze data. If the offset is the same for all gaze samples of the stimulus, then we correct it by shifting them all. When this is not the case, we exclude outlier captured data from the analysis. We end up obtaining a complete data set for 24 out of 37 (71%) participants. This drop-out rate, while high, agrees with the literature for eye-tracking data recorded by fMRI pre-installed eye trackers [176]: it is difficult to avoid noise when conducting fMRI scans and eye-tracking simultaneously.

AOI and Metrics: An *area of interest* (AOI) corresponds to when, and for how long, a subject's eyes focus on a specific area. Following the guidelines of Goldberg and Helfman [111] for defining AOIs in terms of size and granularity, we manually divide every stimulus into four two-dimensional rectangular AOIs: *Pull Request message*, *Code*, *Author Picture*, and *Indicator Image*. The AOI sizes are identical across all stimuli and they are always present on screen.

The *Pull Request message* AOI is provided by the author of the Pull Request to present some information about the proposed code change (i.e., a commit message). The *Code* AOI presents the proposed code changes visually (i.e., as a diff), while the *Author Picture* and *Indicator Image* AOIs display the author of the Pull Request and how to use two fMRI-safe buttons, respectively.

We use the following standard metrics to investigate the impact of provenance on participants' cognitive load and problem-solving strategy. A problem-solving *strategy* models attention distribution and navigation trends over time throughout a task. The *fixation count* indicates the number of attention shifts required to complete the task [109]. Fixation counts often correlate highly with the time spent on a task. The *fixation time* is the total duration of all the fixations on an AOI or the stimulus. Longer fixation time indicates either a relatively high level of interest or difficulty in

[2] http://www.ogama.net/

extracting information and an increased strain on the working memory [114; 113]. The *saccade length* indicates the distance that the eye travels [113]. Larger saccades indicate more meaningful cues while comparing AOI as attention is drawn from a distance [101].

5.4 Results and Analysis

We consider the following research questions:

RQ 5.1 How do the identities of code reviewers and authors change or bias the code review process?

RQ 5.2 Can we classify the gender identities of code reviewers based on patterns of brain activity?

RQ 5.3 Can we differentiate the gender identities of code reviewers based on their visual attention patterns?

RQ 5.4 How do self-reports of the role of identity in code review align with reality?

We make our de-identified dataset (behavioral data, fMRI scan data, eye-tracking data, and survey data) available for analysis and replication at [3].

5.4.1 RQ 5.1 — Behavioral Differences

We examine how code review behaviors (response times and acceptance rates) change as a function of the identities involved using behavioral data from 36 participants.[4]

First, to mitigate false positives, we built a linear mixed effects model (LMM) [304] to investigate the joint effects of Pull Request author and participant identities on response times (RT). Here, we use the notation RT_{A_Woman} to refer to the response time for a Pull Request purportedly authored by a woman, and RT_{P_Man} to refer to the response time for a Pull Request reviewed by

[3] https://web.eecs.umich.edu/ weimerw/fmri.html

[4] One participant did not complete the scan due to physical discomfort.

a man participant. In this model, we treated individual participants as random effects and the authors' and participants' identities as fixed effects. We employed a contrast-based analysis; women participants and machine authors were used as the reference levels (these baselines were chosen by LMM by default and it does not affect the analysis results). We find that both the identities of reviewers (participants) and Pull Request authors have a significant effect on response time: participants' identities: $b = 1.51, SE = 0.77, 95\%CI = [0.02, 3.03], t = 1.97$; authors' identities: $b = -1.14, SE = 0.41, 95\%CI = [-1.96, -0.35], t = -2.759$. Based on the fixed effects results from the linear mixed effect model, we further investigated the relationship between response time and participants' and authors' identities. First, we used Shapiro-Wilk tests to confirm the response time did not follow a normal distribution ($p < 0.001$); we thus used the Mann-Whitney U test to assess the relationship between response times and identities in code review. Our results show that all participants spent significantly less time on Pull Requests that were written by women ($\overline{RT}_{A\_Woman} = 20.8s$, $\overline{RT}_{A\_Man} = 21.7s$, $\overline{RT}_{A\_Machine} = 21.7s$, $p < 0.01$). Furthermore, women reviewers spent significantly less time on all Pull Requests than men ($\overline{RT}_{P\_Woman} = 20.5s$, $\overline{RT}_{P\_Man} = 22.1s$, $p < 0.0001$). Comparing among woman, man and machine author labels, the effect size is large (all $rank - biserial\ r \geq 0.7$).

We also examined the relationship between the acceptance rates and identities using Pearson's Chi-squared Test for significance. We found that machine-written Pull Requests have a lower acceptance rate (78.03%) comparing to man-written (79.68%) and woman-written Pull Requests (84.36%) ($\chi^2(df = 2, n = 1,722) = 8$, $p < 0.05$). The gender bias magnitudes measured here are in line with previous work (e.g., [40]), and on average, human are 4% less likely to accept Pull Requests labeled as written by machine. The effect size of author labels on acceptance rate is small (all $Cramer's\ V < 0.1$) which aligns with observations in previous studies on gender biases in code reviews [40].

Men and women conduct code reviews differently: behaviorally, the gender identity of the reviewer has a significant effect on response time ($p < 0.0001$). Universal biases exist: all

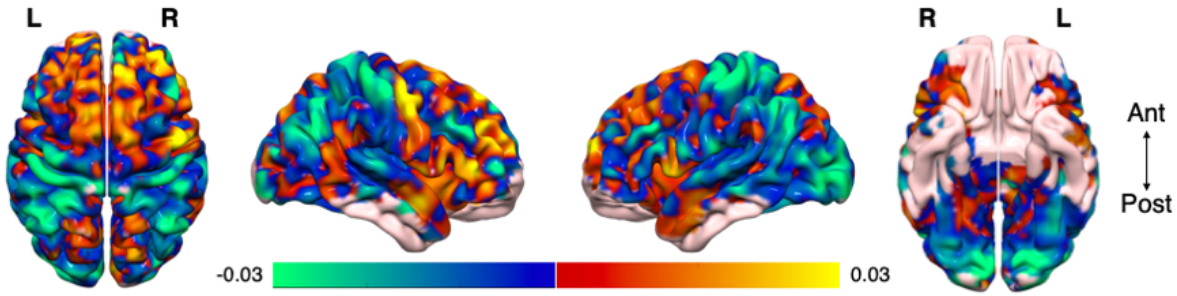


Figure 5.3: Weight map: significant clusters of brain activity for WomanPR > ManPR. When there is stronger activity for woman-authored Pull Requests in 'hot' brain regions, the classifier is pushed towards predicting men participants; more activity in 'cool' brain regions pushes the classifier towards predicting women participants.

participants spend less time evaluating the Pull Requests of women ($t = -2.759$), and all participants are less likely to accept the Pull Requests of machines ($p < 0.05$).

5.4.2 RQ 5.2 — Neurological Differences

We use multivariate pattern classification to determine whether men and women participants exhibit differential neural responses to woman- vs. man-authored Pull Requests (i.e., the contrast in brain activity for $WomanPR > ManPR$). Thirty-six participants' fMRI data is included in this analysis (see Section 5.4.1). Following cross-validation and nonparametric permutation testing, the classifier indeed distinguished between men and women participants significantly better than chance ($BAC = 68.59\%$, $p = 0.016$). This was primarily driven by the ability to accurately identify women participants ($Acc_{Women} = 68.75\%, p = 0.019$); while identification of men participants was similarly-high after cross-validation, accuracy was nonsignificant after permutation testing ($Acc_{Men} = 68.42\%$, $p = 0.527$). A spatial representation of the classifier decision function is shown in Figure 5.3 — note, however, that because these are multivariate weights, localized spatial inferences cannot be made.

Ultimately, these results suggest that — relative to women participants — men show less-consistent differences in their responses to woman- vs. man-authored Pull Requests. That is,

Table 5.2: Pair-wise gender comparisons of eye-gaze data: using non-parametric Wilcoxon Test ($\alpha = 0.05$) for fixation count, fixation time, fixation rate, and saccade length. Significant results ($p < 0.05$) are bolded.

	Mean (Standard Deviation)		
	Women	Men	p
Fixation count	13.0 (13.4)	15.5 (13.8)	**<0.001**
Fixation time (s)	21.6 (7.1)	16.4 (11.5)	0.3
Fixation rate	0.33 (0.34)	0.39 (0.33)	**<0.001**
Saccade length (px)	755.0 (883.1)	561.0 (581.4)	**0.03**

patterns of activity observed in women participants are more similar to one another than men participants are to one another, enabling easier identification of women participants when the model is presented with new data.

It is possible to distinguish women and men conducting code review at a neurological level ($BAC = 68.59\%$, $p = 0.016$). Men and women conduct code reviews differently in terms of associated cognitive processes and patterns of neural activation.

5.4.3 RQ 5.3 — Visual Attention Differences

We analyze eye movements on two levels: globally over the whole stimuli, as well as locally with respect to AOIs. Twenty-four participants' eye-tracking data is included in this analysis (see Section 5.3.2). We measure fixation counts, total fixation times, fixation rates, and saccade lengths over the whole stimulus. The fixation rate is the ratio between fixation count and the total fixation time.

As shown in Table 5.2, we observe a higher level of activity for men participants compared to women. Specifically, men fixated more frequently and made shorter saccades (with regards to the distance traveled) when they were looking at stimuli to evaluate the Pull Request. We also analyze these metrics according to the author's identity (machine, man, or woman) via Friedman tests. No significant effect of author identity was found on these high-level metrics in isolation.

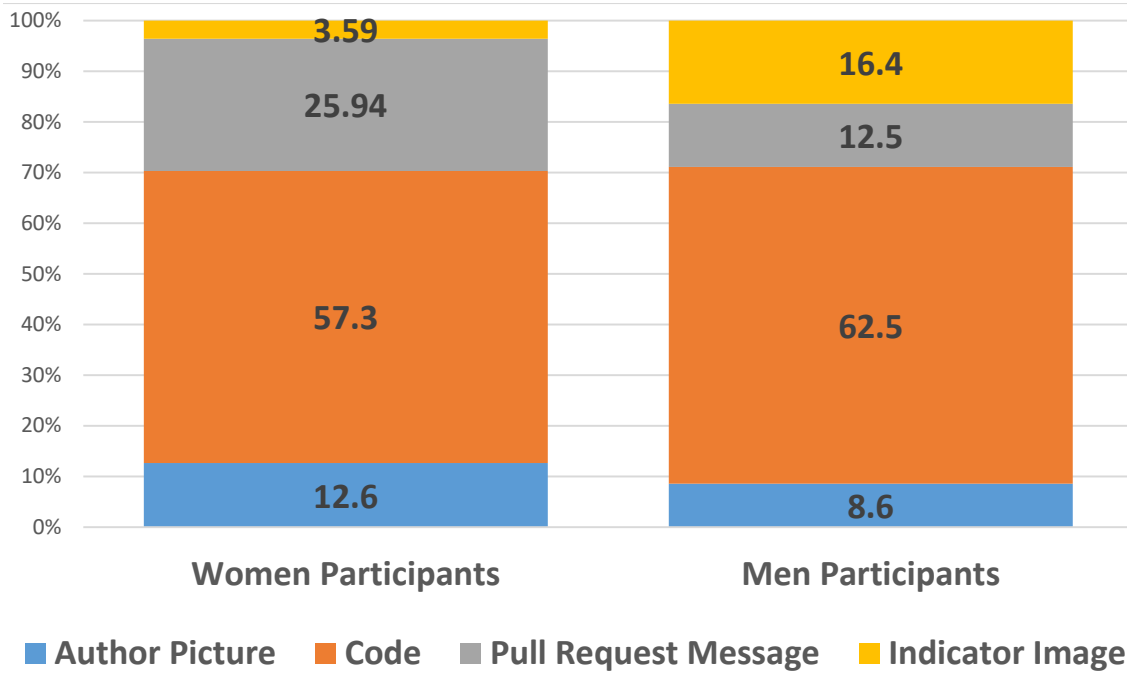


Figure 5.4: Distribution of fixation times across AOIs for men and women participants. Women participants put more attention on reading and processing Pull Request messages and author pictures compared to men.

However, we calculated the metrics mentioned above within each AOI to determine whether a difference exists between the attention distribution of men and women participants while evaluating Pull Requests. We used a general align-and-rank non-parametric factorial analysis [305]. We find that there is a significant interaction between genders: $F(1,3) = 2.64$, $p = 0.05$ for fixation count and $F(1,3) = 4.43$, $p = 0.005$ for fixation time.

Figure 5.4 shows participants' attention distribution across AOIs. Women participants spent significantly more time analyzing Pull Request messages (Wilcoxon test with Bonferroni adjustment: $p < 0.05$) and author picture (Wilcoxon test with Bonferroni adjustment: $p = 0.02$). These results confirm that AOI relevance varies significantly between men and women participants. Specifically, men and women used different patterns of scanning behavior and attention distribution while reviewing code.

(a) A stimulus with a machine author (b) A stimulus with a woman author (c) A stimulus with a man author

Figure 5.5: Heat map: examples of the visual attention in code review: top row for a man participant and bottom row for a woman participant. "Hotter" colors indicate regions with more intense visual attention. More activity is displayed for the men, while women on average spent more time and effort analyzing the Pull Request messages and author pictures.

We summarize a participant's visual attention using a *heat map*. Figure 5.5 displays example heat maps of a man and woman participant analyzing three different stimuli. These heat maps represent visual activity on a color scale — red, orange, green, and blue (warmer to cooler) colors indicate fixation duration. Intuitively, warmer colors indicate locations on the stimulus where a participant focused the most visual attention while evaluating a Pull Request. These heat maps indicate men participants employed a more active scanning pattern (shorter fixation, cooler colors) associated with more frequent attention switching. Additionally, women spent more time and cognitive effort evaluating Pull Request messages and author pictures (regardless of its identity), while men spent more time reading the code. Men and women differ substantially in their visual attention patterns.

Previous work has found that gender differences are likely in problem-solving activities, including programming [306; 307; 308]. Sharafi *et al.* [309] also reported different attention distribution trends based on gender and showed that women participants pay more attention to analyzing and ruling out wrong identifiers. Our results are in broad agreement with the findings of Beckwith *et al.* [306] that men tend to tinker and explore more within an unfamiliar environment and approach the new, unknown features earlier than do women.

Eye-tracking results suggest that men and women participants employ different high-level problem-solving strategies during code review. Men fixated more frequently ($p < 0.001$), while women spent significantly more time analyzing Pull Requests messages and author pictures ($p = 0.02$).

5.4.4 RQ 5.4 — Self-Reporting and Code Review

In our study, all 37 participants provided answers for post-scan questions regarding the tasks and their own experience. To minimize directing participants' self-reports in any particular direction, we employed free response questions. We summarize the six post-survey questions here:

1. What factors do you check (what do you look at, how do you check the content) when you made decisions in code reviews?

2. What were the three most important factors (in order) when you were making decisions in code reviews?

3. How would you compare the machine-generated code changes (i.e., by automated repair tools) with the human-generated changes?

4. Do you think there are any difference between code written by men and women? If there were some, what might they be?

5. Have you observed or thought about any differences between men and women code reviewers?

6. As a software developer, would you be willing to commit machine-generated code into your code base?

We conducted a qualitative analysis of participants' self-report data. The most commonly reported factors in code review that affect participants' decisions were: (1) the quality of comments, (2) whether the description in comments matched code, (3) code readability, and (4) code functionality. These four aspects combined account for 65% of all the reported factors.

Thirty-five of the 37 participants reported they did not notice any difference between the code written by women and men. Only five out of the 37 participants indicated they believed there were behavioral difference between men and women reviewers (e.g., "Women can be more descriptive with the comments", "Perhaps men code reviewers will be more skeptical of code written by women, and women code reviewers will be more cautious in reviewing code written by men").

Only four participants indicated they would consider if a Pull Request was generated by human or machine. However, more participants reported machine-generated Pull Requests in our study to be worse in overall quality, matching intuition, and comments (23 occurrences) than the other

direction (8 occurrences). Indicative quotes from participants are "I think the code generated by machine was more confusing and harder to read. It seemed more complicated than the human-generated code." and "Machine-generated changes are IMO less readable, a little worse in quality, capable in fewer scopes". Without knowing all the Pull Requests and comments were actually written by human programmers, participants expressed negative judgements on those labeled as machine-written. That is, although there were no real differences between the Pull Requests, humans held negative attitudes or biases against machine-generated code. This aligns with the results in Section 5.4.1: humans are less likely to accept Pull Requests generated by machine. Similarly, though the majority of participants reported they believed there was no difference regarding genders of programmers in code reviews, their behaviors displayed significant differences in code reviews (see Section 5.4.1).

Although humans exhibit biases in their acceptance rates of identical code labeled as written by human vs. machines (Section 5.4.1), participant self-reports acknowledge the bias against machines ($23:8$) but do not acknowledge a gender bias. When Pull Request author information changes, participants report seeing quality differences where none exist.

5.4.5 Discussion of Results

Reviewer differences: Our results suggest that men and women conduct code reviews differently. We support this claim with three measurement modalities. Behaviorally, the gender identity of the reviewer has a statistically significant effect on response time. Using medical imaging, we can classify whether neurological data corresponds to a man or woman reviewer. Using eye-tracking, we find that men and women have different attention distributions when reviewing. Note that our results do *not* support any inferences about whether men or women are more accurate at code review. Regardless of the direction of the bias, the code review process overall benefits by identifying and mitigating it [40; 39; 41; 273; 274; 275; 267; 165; 59; 60].

Humans tend to claim no differences between men and women as code reviewers. However,

our results indicate the opposite. Despite no overt behavioral differences (i.e., no significant interaction between participant gender and author identity), the pattern of brain regions recruited when evaluating woman- vs. man-authored code significantly distinguished between men and women participants, with women participants generally showing more reliable patterns of activity (as evidenced by significant classification accuracy for that group). Similarly, our analysis of the distribution of visual attention and the intensity of visual processing reveals that men and women participants have different implicit AOI preferences. While women put more effort into analyzing the pull request messages and author pictures, men fixated more on source code. This finding emphasizes that any a priori assumptions about the importance of different features and various types of information may negatively influence the participants' performance. It may be beneficial to have various sources of information easily accessible to the participants to make an effective judgment without interrupting their train of thought.

In finding statistically-significant differences in how men and women participants carry out software analysis tasks, our results are broadly in line with previous studies (e.g., [309; 306]). We note that a recent medical imaging study of code writing did not find any gender differences [310, Sec. 3.1] but did suggest that code reading and writing are distinct neural tasks.

Author differences: Our results suggest that the contributions of women and machines are not held to the same standards as those of men: they are accepted at different rates and scrutinized for different amounts of time. One null hypothesis is that reviewers are simply correctly favoring better patches (e.g., machine patches may be worse or less maintainable [311; 18]). However, our controlled experiment, in which patch qualities are actually equal, rules out that explanation here. Dual formulations (e.g., women-authored Pull Requests may be of higher quality) are also ruled out by our post-survey data (Section 5.4.4) as well as previous studies [39]. We thus hypothesize that the observed differences result from systematic biases. Such biases have been previously found in software engineering in general and code review in particular [39; 41; 40].

In our study, we observed that humans are 4.7% more likely to accept woman-labeled Pull Requests than man-labeled Pull Requests. Further, they are 4% less likely to accept Pull Re-

quests labeled as machine-generated and humans may hold negative opinions against machine-generated code. These results align with Ryan *et al.*'s findings on trust issues against automated repair tools [281] and other studies on program repair bots [312; 282].

Implications: These neurological and eye-tracking differences do *not* imply inborn biological differences. Indeed, previous fMRI studies on code review using the same classification analysis found such similar differences between experts and novices, regardless of sex [67, Sec. V.3]. This suggests that these observations are more likely attributable to differences in training or feedback. For example, if women are more likely to experience ridicule for failure (e.g., [166; 167; 168; 169; 170; 171; 172]), they may logically adopt different strategies for code review than do men because they perceive different penalties for false positives and false negatives. We view this study as part of a line of work to clarify such biases so that they can be mitigated. For example, follow on work might benefit from investigating which patches, and thus which syntactic or semantic properties of code, were most and least vulnerable to bias (Section 5.4.1). Similarly, if some participants look more at author information (Section 5.4.3), a direct measurement of the reduction in bias that occurs when anonymizing names and author pictures is merited (cf. [41]).

5.5 Threats to Validity

One threat to validity associated with generality is that our selected stimuli may not be indicative. We mitigate this by choosing the Pull Requests randomly from real-world, open-source projects. Similarly, many of our participants are undergraduates. We mitigate this by including a large proportion (30%) of graduate students, and note that, as evaluating the impact of expertise is not the goal of this study, using students as participants is more acceptable [313].

To reduce stereotype threat [314] and social desirability bias [301] and alleviate hypothesis guessing and apprehension, we did not inform the participants about the precise goals of the study. Also, by minimizing the interaction between our team and participants and analyzing de-identified data, we mitigate biases associated with learning or using the identities of individual participants.

Our research team contained both men and women; we conducted a set of pilot studies to help identify biased procedures or results.

To account for conclusion validity, we choose well-documented eye-tracking metrics and analyses [113] as well as well-established and previously-used fMRI analyses [67; 15].

5.6 Chapter Summary

Code review is a critical practice in software engineering. We conducted a study of 37 participants including behavioral, eye-tracking, and medical imaging measurements. Our experiment used historical GitHub Pull Requests but carefully controlled their author information labels, holding quality constant while varying provenance.

We find that men and women conduct code reviews differently in terms of associated visual and cognitive processes and patterns of neural activation. Men and women participants employ different high-level problem-solving strategies during code review: men fixated more frequently ($p < 0.001$), while women spent significantly more time analyzing Pull Request messages and author pictures ($p = 0.02$). Also, the gender of the reviewer has a significant effect on response time ($p < 0.0001$). It is possible to distinguish women and men conducting code review at a neurological level ($BAC = 68.59\%$, $p = 0.016$).

We also find general biases when assessing Pull Requests labeled as written by women or machines. Participants spent less time evaluating the Pull Requests of women ($t = -2.759$), and all participants are less likely to accept the Pull Requests of machines ($p < 0.05$). However, while participant self-reports acknowledge the bias against machines (˜$3\times$), they do not acknowledge a gender bias. When Pull Request author information changes, participants report seeing quality differences where none exist.

We hypothesize that these differences in behaviors and outcomes are related to training and feedback, but more work remains. Our results shed light on potential sources of bias and the physiological mechanisms and behaviors through which they manifest. This chapter presents the

first study to employ both fMRI and eye-tracking to observe potential bias in code review while controlling for quality.

With the ending of this chapter, we have presented all three research components to investigate the cognitive processes in multiple software engineering tasks, including fundamental tasks (i.e. data structure manipulation), higher level tasks (i.e. code writing), as well as the effect of different demographic groups. We presented the application of multiple psycho-physiological measures in these software engineering tasks including fMRI, fNIRS and eye-tracking. The next chapter summaries the three studies presented in this thesis with a look to the future.

CHAPTER 6

Conclusion

Understanding how developers carry out computing activities can help to improve software engineering productivity and guide the use and development of supporting tools and environments. Previous research has explored how programmers conduct computing activities such as code comprehension and code review, but they rely on traditional survey instruments, which may not be reliable. Instead, advances in medical imaging (*i.e.*, fMRI and fNIRS) and eye tracking have recently been applied to software engineering, paving the way for grounded neurobiological understandings of fundamental cognitive processes involved therein.

Using three research components presented in this thesis, we show that it is possible to meaningfully and objectively measure user cognition to understand the role of spatial ability, fundamental processes and stereotypical associations in certain software engineering activities by combining medical imaging and eye tracking:

- In Chapter 3, we investigated our hypothesis that spatial ability is highly associated with software tasks on a foundational level. We exploit two key insights to investigate the relationship between spatial ability and data structure manipulation: (1) the use of multiple medical imaging approaches (*i.e.*, fMRI and fNIRS), and (2) the use of the mental rotation paradigm to serve as a baseline for measuring spatial ability. We designed a controlled experiment using both fMRI and fNIRS (including constructing a customized fNIRS cap) to compare the brain activities of spatial ability (through mental rotation tasks) and data structure tasks. We recruited 76 participants in this work and presented a comparison on costs

and benefits between fMRI and fNIRS for future research in the community.

We found that **data structure and spatial ability operations are similar neurological activities**: both fMRI and fNIRS evidence demonstrates that they involve activations to the same brain regions ($p < 0.01$). However, **some regions relevant to data structures are not accessible to fNIRS**: fNIRS lacked the penetrating power to uncover the full evidence reported by fMRI. We also found that **difficulty matters for data structure tasks**: human brains work even harder for more difficult data structure tasks compared to mental rotations. Though strongly supported by objective measures, this relationship was **not obvious to human participants**, 70% of whom reported no subjective experience of similarity. Our results provide potential for support future research on enhancing programming skills by taking advantage of extant spatial ability training.

- In Chapter 4, we aimed to understand the cognitive processes involved in writing code. We used prose writing as a baseline to ground our results. While some studies have explored how software developers read code, there is no research studying the cognitive processes of creativity in programming such as code writing. We designed and conducted the first fMRI study of code writing and employed a controlled, contrast-based experiment in which 30 participants completed code writing, prose writing, fill-in-the-blank and long response tasks using a customized fMRI-safe keyboard to type their responses in a realistic live editing setting.

  While previous studies have found that code and prose *reading* may be similar at a neurological level, we find that **code and prose writing are quite dissimilar at the neurological level (**$q < 0.05$**)**. At both a low level (i.e., producing a single word or code element) and a high level (i.e., long response coding), we found that writing code requires significantly more activity in brain areas associated with careful, top-down control, planning, and categorization: despite superficial similarity, code appears to be categorically distinct compared to prose. In addition to developing a foundational understanding of code writing, this em-

pirical distinction may be leveraged to develop tools and pedagogies (e.g., transfer training), subsequently potentially affecting large scale workforce retraining and educational reform. Moreover, neurological evidence that code and prose writing are not as intertwined as conventionally thought may encourage more diverse participation in computer science.

- In Chapter 5, we investigated human biases in code review, a critical step in modern software quality assurance. Previous studies have found that software developers do not recognize potential biases when checking the source of code in code reviews and developers may be reluctant to adopt patches generated by automated program repair tools. We conducted a study of 37 participants including behavioral, eye-tracking, and medical imaging measurements to investigate objective sources and characterizations of biases during code review. Our experiment used historical GitHub Pull Requests but carefully controlled their author information labels (man, women, or machine), holding quality constant while varying provenance. We investigated whether the authorship of a Pull Request influences a reviewer's behavior, and whether men and women evaluate Pull Requests differently.

  We found that **men and women conduct code reviews differently in terms of associated visual and cognitive processes and patterns of neural activation**. Men fixated more frequently ($p < 0.001$), while women spent significantly more time analyzing Pull Request messages and author pictures ($p = 0.02$). We also found **general biases when assessing Pull Requests labeled as written by women or machines**. Participants spent less time evaluating the Pull Requests of women ($t = -2.759$), and all participants were less likely to accept the Pull Requests of machines ($p < 0.05$). However, while self-reports acknowledge the bias against machines ($3\times$), they do not acknowledge a gender bias. When Pull Request author information changes, participants report seeing quality differences where none exists. Our results shed light on potential sources of bias and the physiological mechanisms and behaviors through which they manifest. Such an understanding may help design interventions to reduce bias to improve developer productivity.

Table 6.1: Major publications supporting this dissertation

Venue	Title
ICSE'19	Distilling Neural Representations of Data Structure Manipulation using fMRI and fNIRS [15] (Chapter 3)
ICSE'20	Neurological Divide: An fMRI Study of Prose and Code Writing [315] (Chapter 4)
FSE'20	Biases and Differences in Code Review using Medical Imaging and Eye-Tracking: Genders, Humans, and Machines [316] (Chapter 5)
TOSEM'21	Towards an Objective Measure of Developers' Cognitive Activities [317] (Chapter 3)

Table 6.1 lists the major peer-reviewed publications that support the findings presented in this thesis. The work in this thesis addresses the problem of objectively measuring user cognition in software engineering activities using multiple psycho-physiological measures. The work presented in this thesis is among the first to leverage various objective measures to provide a systematic framework to understanding user cognition in programming activities. We presented a systematic approach to study the cognitive processes behind multiple important activities in software engineering that

1. objectively measures relevant factors in computing tasks;
2. is based on rigorous cognitive (neurological and visual) evidence;
3. helps understand semantically-rich and industry-related software engineering activities (e.g, data structure manipulation, code writing and code review);
4. provides guidance for actionable mitigations across different demographic groups.

This approach allowed us to adapt knowledge from other domains (e.g., psychology, biomedical engineering) to design interventions that enhance the effectiveness of modern software engineering and pedagogy techniques. Additionally, this thesis presented guidelines and suggestions for future research on investigating user cognition in software engineering.

6.1 A Look to the Future

While the work presented in this thesis improves our foundational understanding of multiple critical software engineering activities and provides basic principles to adapt psycho-physiological measures to investigate such activities, significant room remains for understanding users' cognitive processes and improving users' productivity in software engineering. Here we summarize future directions regarding the open challenges on this research topic.

We believe investigating user cognition can have high impact on guiding the design of tools, systems and computer science pedagogy. Throughout this thesis, we have addressed the challenges related to our foundational understanding of cognitive processes in software engineering. This paves the way for future work to extend those results and use them as insights for designing effective interventions. Using objective evidence on the neurological and visual level, there is significant room for developing strategies in computing education and workforce training, designing rules for programming tools and systems, as well as support policies for improving diversity and participation in software engineering. While modeling user behavior remains a challenging research topic, the objective measures we describe could lead to a new angle for building a computational model (*e.g.*, with fault tolerance to deal with individual variance). This type of future work could make significant impact on the design paradigms of software engineering. In addition, research on cognition in software engineering may also spark exciting future work with progress in brain computer interaction (BCI).

As of 2021, studies using medical imaging in software engineering struggle with experimental design constraints related to the environment and length of task stimuli. We believe it is important to mitigate these constraints in the future. There are positive news: there are already portable fNIRS machines on the market and researchers in neuroscience and psychology have conducted exploratory work on the application and justification of longer tasks in experimental design. In the future, we expect more research to be done on more complex and realistic software engineering tasks (*e.g.*, more complex software projects, different working environments for multiple users, etc.).

Though psycho-physiological measures, such as medical imaging and eye tracking, are more costly compared to traditional measures, we believe they can be soon more widely adapted with ease to the software engineering community. We have observed that the support for these measures have developed rapidly in the last decade. For example, commercial products for fMRI, fNIRS and eye tracking have been improved over time with lower costs and more technical features. While fMRI requires relatively more professional training for the associated technicians, fNIRS and eye tracking are already common and feasible for researchers with minimum engineering background to adapt to their research activities.

As the research community in software engineering that investigates users' cognitive processes is relatively new, we also believe we need to support replication of our studies. With more research effort on this topic, it is important for us to build a community to share our datasets including experiment protocols, deidentified datasets, and analysis approaches. Since studies in our community commonly involve human participants, we also emphasize the ethical consideration in our research practice.

We firmly believe there are still many interesting and exciting research problems to investigate to better understand user cognition in software engineering. Our work in this thesis paves the way to future explorations along this direction.

BIBLIOGRAPHY

[1] "Gartner says global IT spending to grow 1.1 percent in 2019," retrieved June 1, 2018 from https://www.gartner.com/en/newsroom/press-releases/2019-04-17-gartner-says-global-it-spending-to-grow-1-1-percent-i.

[2] K. Costello and G. Omale, "Gartner says global it spending to reach $3.8 trillion in 2019," in *https://www.gartner.com/en/newsroom/press-releases/2019-01-28-gartner-says-global-it-spending-to-reach--3-8-trillio*, Jan 2019.

[3] N. Singer, "The hard part of computer science? getting into class," in *https://www.nytimes.com/2019/01/24/technology/computer-science-courses-college.html*, Jan 2019.

[4] I. Jacobson, P.-W. Ng, P. E. McMahon, M. Goedicke *et al.*, *The essentials of modern software engineering: free the practices from the method prisons!* Morgan & Claypool, 2019.

[5] C. Ebert, "Open source software in industry," *IEEE Software*, vol. 25, no. 3, pp. 52–53, 2008.

[6] S. Caminiti, "AT&T's $1 billion gambit: Retraining nearly half its workforce for jobs of the future," in *https://www.cnbc.com/2018/03/13/atts-1-billion-gambit-retraining-nearly-half-its-workforce.html*, Mar 2018.

[7] C. Cutter, "Amazon to retrain a third of its U.S. workforce," in *https://www.wsj.com/articles/amazon-to-retrain-a-third-of-its-u-s-workforce-11562841120*, Jul 2019.

[8] F. Steimann, "The paradoxical success of aspect-oriented programming," in *Proceedings of the 21th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2006*, P. L. Tarr and W. R. Cook, Eds. ACM, 2006, pp. 481–497. [Online]. Available: http://doi.acm.org/10.1145/1167473.1167514

[9] R. Pereira and M. Cecília Calani Baranauskas, "A value-oriented and culturally informed approach to the design of interactive systems," *International Journal of Human-Computer Studies*, vol. 80, pp. 66–82, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1071581915000592

[10] D. Aharoni, "What you see is what you get the influence of visualization on the perception of data structures," *DOCUMENT RESUME*, vol. 11, no. 4, p. 10, 2000.

[11] R. D. Pea and D. M. Kurland, "On the cognitive effects of learning computer programming," *New ideas in psychology*, vol. 2, no. 2, pp. 137–168, 1984.

[12] R. Brooks, "Towards a theory of the cognitive processes in computer programming," *International Journal of Man-Machine Studies*, vol. 9, no. 6, pp. 737–751, 1977.

[13] A. Eteläpelto, "Metacognition and the expertise of computer program comprehension," *Scandinavian Journal of Educational Research*, vol. 37, no. 3, pp. 243–254, 1993.

[14] Q. Fan, "The effects of beacons, comments, and tasks on program comprehension process in software maintenance," Ph.D. dissertation, University of Maryland, Baltimore County, Catonsville, MD, USA, 2010, aAI3422807.

[15] Y. Huang, X. Liu, R. Krueger, T. Santander, X. Hu, K. Leach, and W. Weimer, "Distilling neural representations of data structure manipulation using fMRI and fNIRS," in *International Conference on Software Engineering (ICSE)*, 2019.

[16] S. Riley, "Password security: What users know and what they actually do," *Usability News*, vol. 8, no. 1, pp. 2833–2836, 2006.

[17] N. Dell, V. Vaidyanathan, I. Medhi, E. Cutrell, and W. Thies, "Yours is better!: participant response bias in HCI," in *Human Factors in Computing Systems*. ACM, 2012, pp. 1321–1330.

[18] Z. P. Fry, B. Landau, and W. Weimer, "A human study of patch maintainability," in *International Symposium on Software Testing and Analysis*, 2012, pp. 177–187.

[19] A. Vashistha, F. Okeke, R. Anderson, and N. Dell, "'you can always do better!" the impact of social proof on participant response bias," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–13.

[20] T. W. Robbins, "Cognition: the ultimate brain function," *Neuropsychopharmacology*, vol. 36, no. 1, pp. 1–2, 2011.

[21] M. Hegarty and M. Kozhevnikov, "Types of visual–spatial representations and mathematical problem solving." *Journal of educational psychology*, vol. 91, no. 4, p. 684, 1999.

[22] J. Wai, D. Lubinski, and C. P. Benbow, "Spatial ability for stem domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance." *Journal of Educational Psychology*, vol. 101, no. 4, p. 817, 2009.

[23] D. M. Tracy, "Toys, spatial ability, and science and mathematics achievement: Are they related?" *Sex roles*, vol. 17, no. 3-4, pp. 115–138, 1987.

[24] E.-M. Yang, T. Andre, T. J. Greenbowe, and L. Tibell, "Spatial ability and the impact of visualization/animation on learning electrochemistry," *International Journal of Science Education*, vol. 25, no. 3, pp. 329–349, 2003.

[25] M. Alias, T. R. Black, and D. E. Gray, "Effect of instruction on spatial visualization ability in civil engineering students," *International Education Journal*, vol. 3, no. 1, 2002.

[26] A. A. Black, "Spatial ability and earth science conceptual understanding," *Journal of Geoscience Education*, vol. 53, no. 4, pp. 402–414, 2005.

[27] S. D. Moffat, E. Hampson, and M. Hatzipantelis, "Navigation in a "virtual" maze: Sex differences and correlation with psychometric measures of spatial ability in humans," *Evolution and Human Behavior*, vol. 19, no. 2, pp. 73–87, 1998.

[28] C. J. Limb and A. R. Braun, "Neural substrates of spontaneous musical performance: An fMRI study of jazz improvisation," *PLoS one*, vol. 3, no. 2, p. e1679, 2008.

[29] M. L. Pelchat, A. Johnson, R. Chan, J. Valdez, and J. D. Ragland, "Images of desire: food-craving activation during fMRI," *Neuroimage*, vol. 23, no. 4, pp. 1486–1493, 2004.

[30] V. W. Berninger and W. Winn, "Implications of advancements in brain research and technology for writing development, writing instruction, and educational evolution," *Handbook of writing research*, pp. 96–114, 2006.

[31] V. Menon and J. Desmond, "Left superior parietal cortex involvement in writing: integrating fMRI with lesion evidence," *Cognitive brain research*, vol. 12, no. 2, pp. 337–340, 2001.

[32] S. Planton, M. Jucla, F.-E. Roux, and J.-F. Démonet, "The "handwriting brain": a meta-analysis of neuroimaging studies of motor versus orthographic processes," *Cortex*, vol. 49, no. 10, pp. 2772–2787, 2013.

[33] R. D. Pea and D. M. Kurland, "On the cognitive prerequisites of learning computer programming," Center for Children and Technology, Tech. Rep. 18, 1983.

[34] R. Bednarik and M. Tukiainen, "An eye-tracking methodology for characterizing program comprehension processes," in *Proceedings of the 2006 symposium on Eye tracking research & applications*. ACM, 2006, pp. 125–132.

[35] J. Siegmund, C. Kästner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann, "Understanding understanding source code with functional magnetic resonance imaging," in *International Conference on Software Engineering*, 2014, pp. 378–389.

[36] J. Castelhano, I. C. Duarte, C. Ferreira, J. Duraes, H. Madeira, and M. Castelo-Branco, "The Role of the Insula in Intuitive Expert Bug Detection in Computer Code: An fMRI Study," *Brain Imaging and Behavior*, May 2018.

[37] J. Duraes, H. Madeira, J. Castelhano, C. Duarte, and M. C. Branco, "WAP: Understanding the Brain at Software Debugging," in *International Symposium on Software Reliability Engineering*, 2016, pp. 87–92.

[38] S. Fakhoury, Y. Ma, V. Arnaoudova, and O. Adesope, "The effect of poor source code lexicon and readability on developers' cognitive load," in *International Conference on Program Comprehension*, 2018.

[39] N. Imtiaz, J. Middleton, J. Chakraborty, N. Robson, G. Bai, and E. R. Murphy-Hill, "Investigating the effects of gender bias on GitHub," in *International Conference on Software Engineering (ICSE)*, 2019, pp. 700–711. [Online]. Available: https://doi.org/10.1109/ICSE.2019.00079

[40] J. Terrell, A. Kofink, J. Middleton, C. Rainear, E. Murphy-Hill, C. Parnin, and J. Stallings, "Gender differences and bias in open source: Pull request acceptance of women versus men," *PeerJ Computer Science*, vol. 3, p. e111, 2017.

[41] D. Ford, M. Behroozi, A. Serebrenik, and C. Parnin, "Beyond the code itself: how programmers really look at pull requests," in *International Conference on Software Engineering: Software Engineering in Society*, 2019.

[42] B. De Smedt, D. Ansari, R. H. Grabner, M. M. Hannula, M. Schneider, and L. Verschaffel, "Cognitive neuroscience meets mathematics education." *Educational Research Review*, vol. 5, no. 1, pp. 97–105, 2010.

[43] S. J. Pickering and P. Howard-Jones, "Educators' views on the role of neuroscience in education: Findings from a study of uk and international perspectives," *Mind, Brain, and Education*, vol. 1, no. 3, pp. 109–113, 2007.

[44] M. Atherton, J. Zhuang, W. M. Bart, X. Hu, and S. He, "A functional MRI study of high-level cognition. I. the game of chess," *Cognitive Brain Research*, vol. 16, no. 1, pp. 26–31, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926641002002070

[45] J. S. Ross, J. Tkach, P. M. Ruggieri, M. Lieber, and E. Lapresto, "The mind's eye: Functional MR imaging evaluation of golf motor imagery," *American Journal of Neuroradiology*, vol. 24, no. 6, pp. 1036–1044, 2003. [Online]. Available: http://www.ajnr.org/content/24/6/1036.abstract

[46] E. A. Maguire, K. Woollett, and H. J. Spiers, "London taxi drivers and bus drivers: A structural MRI and neuropsychological analysis," *Hippocampus*, vol. 16, no. 12, pp. 1091–1101, 2006. [Online]. Available: http://dx.doi.org/10.1002/hipo.20233

[47] M. P. Milham, K. I. Erickson, M. T. Banich, A. F. Kramer, A. Webb, T. Wszalek, and N. J. Cohen, "Attentional control in the aging brain: Insights from an fMRI study of the Stroop task," *Brain and Cognition*, vol. 49, no. 3, pp. 277–296, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0278262601915015

[48] R. Cabeza, S. M. Daselaar, F. Dolcos, S. E. Prince, M. Budde, and L. Nyberg, "Task-independent and task-specific age effects on brain activity during working memory, visual attention and episodic retrieval," *Cerebral Cortex*, vol. 14, no. 4, pp. 364–375, 2004. [Online]. Available: http://cercor.oxfordjournals.org/content/14/4/364.abstract

[49] M. Ritchey, A. P. Yonelinas, and C. Ranganath, "Functional connectivity relationships predict similarities in task activation and pattern information during associative memory encoding," *J. Cognitive Neuroscience*, vol. 26, no. 5, pp. 1085–1099, May 2014. [Online]. Available: http://dx.doi.org/10.1162/jocn_a_00533

[50] D. S. Bassett, N. F. Wymbs, M. A. Porter, P. J. Mucha, J. M. Carlson, and S. T. Grafton, "Dynamic reconfiguration of human brain networks during learning," *Proceedings of the National Academy of Sciences*, vol. 108, no. 18, pp. 7641–7646, 2011. [Online]. Available: http://www.pnas.org/content/108/18/7641.abstract

[51] P. Mabe and S. West, "Validity of self-evaluation of ability: A review and meta-analysis," *Journal of Applied Psychology*, vol. 67, no. 3, pp. 280–296, 6 1982.

[52] P. M. Podsakoff and D. W. Organ, "Self-reports in organizational research: Problems and prospects," *Journal of Management*, vol. 12, no. 4, pp. 531–544, 1986. [Online]. Available: http://jom.sagepub.com/content/12/4/531.abstract

[53] D. R. Leff, F. Orihuela-Espina, C. E. Elwell, T. Athanasiou, D. T. Delpy, A. W. Darzi, and G.-Z. Yang, "Assessment of the cerebral cortex during motor task behaviours in adults: a systematic review of functional near infrared spectroscopy (fNIRS) studies," *Neuroimage*, vol. 54, no. 4, pp. 2922–2936, 2011.

[54] M. A. Lindquist, J. M. Loh, L. Y. Atlas, and T. D. Wager, "Modeling the hemodynamic response function in fMRI: efficiency, bias and mis-modeling," *Neuroimage*, vol. 45, no. 1, pp. S187–S198, 2009.

[55] M. Fagan, "Design and code inspections to reduce errors in program development," in *Software pioneers*. Springer, 2002, pp. 575–607.

[56] A. Bacchelli and C. Bird, "Expectations, outcomes, and challenges of modern code review," in *Proceedings of the 2013 international conference on software engineering*. IEEE Press, 2013, pp. 712–721.

[57] C. Jones, "Measuring defect potentials and defect removal efficiency," *CrossTalk The Journal of Defense Software Engineering*, vol. 21, no. 6, pp. 11–13, 2008.

[58] J. Cohen, E. Brown, B. DuRette, and S. Teleki, *Best kept secrets of peer code review*. Smart Bear Somerville, 2006.

[59] V. Pieterse, D. G. Kourie, and I. P. Sonnekus, "Software engineering team diversity and performance," in *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*. South African Institute for Computer Scientists and Information Technologists, 2006, pp. 180–186.

[60] L. F. Capretz and F. Ahmed, "Why do we need personality diversity in software engineering?" *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 2, pp. 1–11, 2010.

[61] S. Rangarajan, "Here's the clearest picture of Silicon Valley's diversity yet: It's bad. But some companies are doing less bad," https://www.revealnews.org/article/heres-the-clearest-picture-of-silicon-valleys-diversity-yet/, 2018, [Online; accessed 4-September-2019].

[62] L. Holloway, "4 Big Things Google Is Doing To Improve Diversity In Silicon Valley," https://www.revealnews.org/article/heres-the-clearest-picture-of-silicon-valleys-diversity-yet/, 2015, [Online; accessed 4-September-2019].

[63] S. A. Sorby and B. J. Baartmans, "The development and assessment of a course for enhancing the 3-d spatial visualization skills of first year engineering students," *Journal of Engineering Education*, vol. 89, no. 3, pp. 301–307, 2000.

[64] S. A. Sorby, "Educational research in developing 3-d spatial skills for engineering students," *International Journal of Science Education*, vol. 31, no. 3, pp. 459–480, 2009.

[65] N. Hoyek, C. Collet, O. Rastello, P. Fargier, P. Thiriet, and A. Guillot, "Enhancement of mental rotation abilities and its effect on anatomy learning," *Teaching and learning in medicine*, vol. 21, no. 3, pp. 201–206, 2009.

[66] F. H. Rauscher, G. L. Shaw, and K. N. Ky, "Listening to mozart enhances spatial-temporal reasoning: towards a neurophysiological basis," *Neuroscience letters*, vol. 185, no. 1, pp. 44–47, 1995.

[67] B. Floyd, T. Santander, and W. Weimer, "Decoding the representation of code in the brain: An fMRI study of code review and expertise," in *International Conference on Software Engineering*, 2017, pp. 175–186.

[68] "Blood-oxygen-level-dependent imaging," retrieved March 15, 2021 from https://en.wikipedia.org/wiki/Blood-oxygen-level-dependent_imaging.

[69] "What is neuroimaging," retrieved March 15, 2021 from https://medicine.utah.edu/psychiatry/research/labs/diagnostic-neuroimaging/neuroimaging.php.

[70] R. B. Buxton, K. Uludağ, D. J. Dubowitz, and T. T. Liu, "Modeling the hemodynamic response to brain activation," *Neuroimage*, vol. 23, pp. S220–S233, 2004.

[71] L. J. Garey, *Brodmann's "Localisation in the Cerebral Cortex"*. World Scientific, 2006.

[72] S. Ogawa, T.-M. Lee, A. R. Kay, and D. W. Tank, "Brain magnetic resonance imaging with contrast dependent on blood oxygenation," *Proceedings of the National Academy of Sciences*, vol. 87, no. 24, pp. 9868–9872, 1990.

[73] R. H. Hashemi, W. G. Bradley, and C. J. Lisanti, *MRI: the basics: The Basics*. Lippincott Williams & Wilkins, 2012.

[74] S. Ulmer and O. Jansen, *fMRI*. Springer, 2010.

[75] Y. Ozaki and S. Kawata, "Near-infrared spectroscopy," *Gakkai Shuppan Center, Tokyo, Japan*, 1996.

[76] B. P. Acevedo, E. N. Aron, A. Aron, M.-D. Sangster, N. Collins, and L. L. Brown, "The highly sensitive brain: an fMRI study of sensory processing sensitivity and response to others' emotions," *Brain and behavior*, vol. 4, no. 4, pp. 580–594, 2014.

[77] W. D. Gaillard, B. C. Sachs, J. R. Whitnah, Z. Ahmad, L. M. Balsamo, J. R. Petrella, S. H. Braniecki, C. M. McKinney, K. Hunter, B. Xu *et al.*, "Developmental aspects of language processing: fMRI of verbal fluency in children and adults," *Human brain mapping*, vol. 18, no. 3, pp. 176–185, 2003.

[78] C. J. Stoodley, E. M. Valera, and J. D. Schmahmann, "Functional topography of the cerebellum for motor and cognitive tasks: an fMRI study," *Neuroimage*, vol. 59, no. 2, pp. 1560–1570, 2012.

[79] M. Okamoto, M. Matsunami, H. Dan, T. Kohata, K. Kohyama, and I. Dan, "Prefrontal activity during taste encoding: an fNIRS study," *Neuroimage*, vol. 31, no. 2, pp. 796–806, 2006.

[80] L.-A. Petitto, M. S. Berens, I. Kovelman, M. H. Dubins, K. Jasinska, and M. Shalinsky, "The "perceptual wedge hypothesis" as the basis for bilingual babies' phonetic processing advantage: New insights from fNIRS brain imaging," *Brain and language*, vol. 121, no. 2, pp. 130–143, 2012.

[81] J. Liu, A. Harris, and N. Kanwisher, "Perception of face parts and face configurations: an fMRI study," *Journal of cognitive neuroscience*, vol. 22, no. 1, pp. 203–211, 2010.

[82] J. P. O'Doherty, A. Hampton, and H. Kim, "Model-based fMRI and its application to reward learning and decision making," *Annals of the New York Academy of sciences*, vol. 1104, no. 1, pp. 35–53, 2007.

[83] M. P. Van Den Heuvel and H. E. H. Pol, "Exploring the brain network: a review on resting-state fMRI functional connectivity," *European neuropsychopharmacology*, vol. 20, no. 8, pp. 519–534, 2010.

[84] S. M. Smith, P. T. Fox, K. L. Miller, D. C. Glahn, P. M. Fox, C. E. Mackay, N. Filippini, K. E. Watkins, R. Toro, A. R. Laird *et al.*, "Correspondence of the brain's functional architecture during activation and rest," *Proceedings of the National Academy of Sciences*, vol. 106, no. 31, pp. 13 040–13 045, 2009.

[85] M. M. Monti, A. Vanhaudenhuyse, M. R. Coleman, M. Boly, J. D. Pickard, L. Tshibanda, A. M. Owen, and S. Laureys, "Willful modulation of brain activity in disorders of consciousness," *New England Journal of Medicine*, vol. 362, no. 7, pp. 579–589, 2010.

[86] K. Smith, "fMRI 2.0," *Nature*, vol. 484, no. 7392, p. 24, 2012.

[87] D. A. Boas, C. E. Elwell, M. Ferrari, and G. Taga, "Twenty years of functional near-infrared spectroscopy: introduction for the special issue," 2014.

[88] S. Lloyd-Fox, A. Blasi, and C. Elwell, "Illuminating the developing brain: the past, present and future of functional near infrared spectroscopy," *Neuroscience & Biobehavioral Reviews*, vol. 34, no. 3, pp. 269–284, 2010.

[89] A.-C. Ehlis, S. Schneider, T. Dresler, and A. J. Fallgatter, "Application of functional near-infrared spectroscopy in psychiatry," *Neuroimage*, vol. 85, pp. 478–488, 2014.

[90] H. Obrig, "NIRS in clinical neurology — a 'promising' tool?" *Neuroimage*, vol. 85, pp. 535–546, 2014.

[91] R. N. Henson, C. J. Price, M. D. Rugg, R. Turner, and K. J. Friston, "Detecting latency differences in event-related bold responses: application to words versus nonwords and initial versus repeated face presentations," *Neuroimage*, vol. 15, no. 1, pp. 83–97, 2002.

[92] R. Aamand, T. Dalsgaard, Y.-C. Lynn Ho, A. Moller, A. Roepstorff, and T. Lund, "A NO way to BOLD?: Dietary nitrate alters the hemodynamic response to visual stimulation," *NeuroImage*, vol. 83, 07 2013.

[93] Scicurious, "IgNobel prize in neuroscience: The dead salmon study," *Scientific American Blog Network*, Sep 2012. [Online]. Available: https://blogs.scientificamerican.com/scicurious-brain/ignobel-prize-in-neuroscience-the-dead-salmon-study/

[94] C. M. Bennett, M. Miller, and G. Wolford, "Neural correlates of interspecies perspective taking in the post-mortem atlantic salmon: an argument for multiple comparisons correction," *Neuroimage*, vol. 47, no. Suppl 1, p. S125, 2009.

[95] E. Amaro Jr and G. J. Barker, "Study design in fmri: basic principles," *Brain and cognition*, vol. 60, no. 3, pp. 220–232, 2006.

[96] "Functional magnetic resonance imaging," retrieved March 15, 2021 from https://en.wikipedia.org/wiki/Functional_magnetic_resonance_imaging.

[97] G. Aguirre and M. D'Esposito, "Experimental design for brain fmri," *Functional MRI*, pp. 369–380, 1999.

[98] T. J. Grabowski and A. R. Damasio, "Investigating language with functional neuroimaging," in *Brain mapping: The systems*. Elsevier, 2000, pp. 425–461.

[99] F. Hermens, R. Flin, and I. Ahmed, "Eye movements in surgery: A literature review," *Journal of Eye Movement Research*, vol. 6, no. 4, Nov. 2013. [Online]. Available: https://bop.unibe.ch/JEMR/article/view/2363

[100] Z. Zhang and J. Zhang, "A new real-time eye tracking based on nonlinear unscented kalman filter for monitoring driver fatigue," *Journal of Control Theory and Applications*, vol. 8, no. 2, pp. 181–188, 2010.

[101] A. Poole and L. J. Ball, "Eye tracking in human-computer interaction and usability research: Current status and future," in *Prospects", Chapter in C. Ghaoui (Ed.): Encyclopedia of Human-Computer Interaction. Pennsylvania: Idea Group, Inc*. Hershey, PA: Information Science Reference - Imprint of: IGI Publishing, Dec 2005, pp. 1–5.

[102] V. Sundstedt, "Gazing at games: Using eye tracking to control virtual characters," in *ACM SIGGRAPH 2010 Courses*, ser. SIGGRAPH '10. New York, NY, USA: ACM, 2010, pp. 5:1–5:160. [Online]. Available: http://doi.acm.org/10.1145/1837101.1837106

[103] S. Alkan and K. Cagiltay, "Studying computer game learning experience through eye tracking," *British Journal of Educational Technology*, vol. 38, no. 3, pp. 538–542, 2007.

[104] U. Obaidellah, M. Al Haek, and P. C.-H. Cheng, "A survey on the usage of eye-tracking in computer programming," *ACM Comput. Surv.*, vol. 51, no. 1, pp. 5:1–5:58, Jan. 2018. [Online]. Available: http://doi.acm.org/10.1145/3145904

[105] Z. Sharafi, Z. Soh, and Y.-G. Guéhéneuc, "A systematic literature review on the usage of eye-tracking in software engineering," *Inf. Softw. Technol.*, vol. 67, no. C, p. 79–107, Nov. 2015. [Online]. Available: https://doi-org.proxy.lib.umich.edu/10.1016/j.infsof.2015.06.008

[106] K. Rayner, "Eye movements in reading and information processing," *Psychological Bulletin*, vol. 85, no. 3, pp. 618–660, 1978.

[107] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *International Journal of Industrial Ergonomics*, vol. 24, no. 6, pp. 631–645, 1999.

[108] A. Duchowski, *Eye tracking methodology: Theory and practice*. Springer-Verlag New York Inc, 2007.

[109] M. A. Just and P. A. Carpenter, "A theory of reading: from eye fixations to comprehension." *Psychological review*, vol. 87, no. 4, p. 329, 1980.

[110] R. Bednarik, "Expertise-dependent visual attention strategies develop over time during debugging with multiple code representations," *International Journal of Human-Computer Studies*, vol. 70, no. 2, pp. 143–155, Feb. 2012.

[111] J. H. Goldberg and J. I. Helfman, "Comparing information graphics: A critical look at eye tracking," in *Proceedings of the 3rd BEyond Time and Errors: Novel evaLuation Methods for Information Visualization Workshop*, ser. BELIV '10. New York, NY, USA: ACM, 2010, pp. 71–78. [Online]. Available: http://doi.acm.org/10.1145/2110192.2110203

[112] Z. Sharafi, B. Sharif, Y.-G. Guéhéneuc, A. Begel, R. Bednarik, and M. Crosby, "A practical guide on conducting eye tracking studies in software engineering," *Empirical Software Engineering*, pp. 1–47, 2020.

[113] Z. Sharafi, T. Shaffer, B. Sharif, and Y.-G. Guéhéneuc, "Eye-tracking metrics in software engineering," in *Proceeding of 2015 Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 2015, pp. 96–103.

[114] R. J. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," *Mind*, vol. 2, no. 3, p. 4, 2003.

[115] T. Donnon, J.-G. DesCôteaux, and C. Violato, "Impact of cognitive imaging and sex differences on the development of laparoscopic suturing skills," *Canadian journal of surgery*, vol. 48, no. 5, p. 387, 2005.

[116] M. C. Corballis, "Mental rotation and the right hemisphere," *Brain and language*, vol. 57, no. 1, pp. 100–121, 1997.

[117] I. M. Harris, G. F. Egan, C. Sonkkila, H. J. Tochon-Danguy, G. Paxinos, and J. D. Watson, "Selective right parietal lobe activation during mental rotation: a parametric pet study," *Brain*, vol. 123, no. 1, pp. 65–73, 2000.

[118] J. C. Culham and N. G. Kanwisher, "Neuroimaging of cognitive functions in human parietal cortex," *Current opinion in neurobiology*, vol. 11, no. 2, pp. 157–163, 2001.

[119] M. S. Cohen, S. M. Kosslyn, H. C. Breiter, G. J. DiGirolamo, W. L. Thompson, A. Anderson, S. Bookheimer, B. R. Rosen, and J. Belliveau, "Changes in cortical activity during mental rotation a mapping study using functional mri," *Brain*, vol. 119, no. 1, pp. 89–100, 1996.

[120] R. N. Shepard and J. Metzler, "Mental rotation of three-dimensional objects," *Science*, vol. 171, no. 3972, pp. 701–703, 1971.

[121] A. Gogos, M. Gavrilescu, S. Davison, K. Searle, J. Adams, S. L. Rossell, R. Bell, S. R. Davis, and G. F. Egan, "Greater superior than inferior parietal lobule activation with increasing rotation angle during mental rotation: an fMRI study," *Neuropsychologia*, vol. 48, no. 2, pp. 529–535, 2010.

[122] "Prose writing," retrieved March 15, 2021 from https://literarydevices.net/prose/.

[123] L. Flower and J. R. Hayes, "A cognitive process theory of writing," *College composition and communication*, vol. 32, no. 4, pp. 365–387, 1981.

[124] A. Cumming, "Writing expertise and second-language proficiency," *Language learning*, vol. 39, no. 1, pp. 81–135, 1989.

[125] I. Leki, "Building expertise through sequenced writing assignments," *Teachers of English to Speakers of Other Languages Journal*, vol. 1, no. 2, pp. 19–23, 1992.

[126] A. Beaufort, "Learning the trade: A social apprenticeship model for gaining writing expertise," *Written communication*, vol. 17, no. 2, pp. 185–223, 2000.

[127] C. Shah, K. Erhard, H.-J. Ortheil, E. Kaza, C. Kessler, and M. Lotze, "Neural correlates of creative writing: an fMRI study," *Human brain mapping*, vol. 34, no. 5, pp. 1088–1101, 2013.

[128] G. Sugihara, T. Kaminaga, and M. Sugishita, "Interindividual uniformity and variety of the "writing center": a functional MRI study," *Neuroimage*, vol. 32, no. 4, pp. 1837–1849, 2006.

[129] S. Planton, M. Longcamp, P. Péran, J.-F. Demonet, and M. Jucla, "How specialized are writing-specific brain regions? an fMRI study of writing, drawing and oral spelling," *Cortex*, vol. 88, pp. 66–80, 2017.

[130] J. J. Purcell, E. M. Napoliello, and G. F. Eden, "A combined fMRI study of typed spelling and reading," *NeuroImage*, vol. 55, no. 2, pp. 750 – 762, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S105381191001520X

[131] S. R. Hooper, L.-J. C. Costa, M. McBee, K. L. Anderson, D. C. Yerby, A. Childress, and S. B. Knuth, "A written language intervention for at-risk second grade students: a randomized controlled trial of the process assessment of the learner lesson plans in a tier 2 response-to-intervention (RtI) model," *Annals of dyslexia*, vol. 63, no. 1, pp. 44–64, 2013.

[132] R. T. Kellogg and A. P. Whiteford, "Training advanced writing skills: The case for deliberate practice," *Educational Psychologist*, vol. 44, no. 4, pp. 250–266, 2009.

[133] C. Titz and J. Karbach, "Working memory and executive functions: effects of training on academic achievement," *Psychological research*, vol. 78, no. 6, pp. 852–868, 2014.

[134] D. Aharoni, "Cogito, ergo sum! cognitive processes of students dealing with data structures," *ACM SIGCSE Bulletin*, vol. 32, no. 1, pp. 26–30, 2000.

[135] N. J. Haneef, "Software documentation and readability: a proposed process improvement," *SIGSOFT Softw. Eng. Notes*, vol. 23, no. 3, pp. 75–77, 1998.

[136] D. E. Knuth, "Literate programming," *Comput. J.*, vol. 27, no. 2, pp. 97–111, 1984.

[137] J. C. Knight and E. A. Myers, "An improved inspection technique," *Commun. ACM*, vol. 36, no. 11, pp. 51–61, Nov. 1993.

[138] F. Shull, I. Rus, and V. Basili, "Improving software inspections by using reading techniques," in *Proceedings of the 23rd International Conference on Software Engineering*, ser. ICSE '01. USA: IEEE Computer Society, 2001, p. 726–727.

[139] J. L. Elshoff and M. Marcotty, "Improving computer program readability to aid modification," *Commun. ACM*, vol. 25, no. 8, pp. 512–521, 1982.

[140] R. P. L. Buse and T. Zimmermann, "Information needs for software development analytics," in *International Conference on Software Engineering*, 2012, pp. 987–996.

[141] D. E. Knuth, *The art of computer programming*. Pearson Education, 1997, vol. 3.

[142] S. Maguire, *Writing solid code*. Greyden Press, LLC, 2013.

[143] R. C. Martin, *Clean code: a handbook of agile software craftsmanship*. Pearson Education, 2009.

[144] M. Fowler, *Refactoring: improving the design of existing code*. Addison-Wesley Professional, 2018.

[145] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Scratch: Programming for all," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, Nov. 2009.

[146] D. C. Smith, "Pygmalion: a creative programming environment," STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, Tech. Rep., 1975.

[147] J. S. Bruner, J. J. Goodnow, and G. A. Austin, *A study of thinking*. New York: John Wiley & Sons, Inc, 1956.

[148] M. Sime, T. Green, and D. Guest, "Psychological evaluation of two conditional constructions used in computer languages," *International Journal of Man-Machine Studies*, vol. 5, no. 1, pp. 105–113, 1973.

[149] E. E. Grant and H. Sackman, "An exploratory investigation of programmer performance under on-line and off-line conditions," *IEEE Transactions on Human Factors in Electronics*, no. 1, pp. 33–48, 1967.

[150] E. A. Youngs, "Human errors in programming," *International Journal of Man-Machine Studies*, vol. 6, no. 3, pp. 361–376, 1974.

[151] M. Weiser and J. Shertz, "Programming problem representation in novice and expert programmers," *International Journal of Man-Machine Studies*, vol. 19, no. 4, pp. 391–398, 1983.

[152] T. Ormerod, "Human cognition and programming," in *Psychology of programming*. Elsevier, 1990, pp. 63–82.

[153] J. Brandt, P. J. Guo, J. Lewenstein, M. Dontcheva, and S. R. Klemmer, "Two studies of opportunistic programming: interleaving web foraging, learning, and writing code," in *Conference on Human Factors in Computing Systems*, 2009, pp. 1589–1598.

[154] I. Sommerville, *Software Engineering*, 9th ed. Pearson, 2010, vol. 137035152.

[155] N. Kennedy, "Google Mondrian: web-based code review and storage," 2006, [Online; accessed February-2020].

[156] A. Tsotsis, "Meet Phabricator, the witty code review tool built inside Facebook," 2011, [Online; accessed February-2020].

[157] D. Huizinga and A. Kolawa, *Automated Defect Prevention: Best Practices in Software Management*, 1st ed. USA: Wiley, 2007.

[158] A. F. Ackerman, L. S. Buchwald, and F. H. Lewski, "Software inspections: An effective verification process," *IEEE Softw.*, vol. 6, no. 3, pp. 31–36, May 1989. [Online]. Available: http://dx.doi.org/10.1109/52.28121

[159] A. Dunsmore, M. Roper, and M. Wood, "Practical code inspection techniques for object-oriented systems: An experimental comparison," *IEEE Softw.*, vol. 20, no. 4, pp. 21–29, Jul. 2003. [Online]. Available: http://dx.doi.org/10.1109/MS.2003.1207450

[160] M. E. Fagan, "Design and code inspections to reduce errors in program development," *IBM Syst. J.*, vol. 38, no. 2-3, pp. 258–287, Jun. 1999. [Online]. Available: http://dx.doi.org/10.1147/sj.382.0258

[161] L. Williamson, "IBM Rational software analyzer: Beyond source code," in *Rational Software Developer Conference*, Florida, USA, 07 2008.

[162] C. Weiß, R. Premraj, T. Zimmermann, and A. Zeller, "How long will it take to fix this bug?" in *Workshop on Mining Software Repositories*. Minneapolis, MN, USA: IEEE, May 2007, pp. 1–1.

[163] Z. Yin, D. Yuan, Y. Zhou, S. Pasupathy, and L. Bairavasundaram, "How do fixes become bugs?" in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, ser. ESEC/FSE '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 26–36. [Online]. Available: https://doi-org.proxy.lib.umich.edu/10.1145/2025113.2025121

[164] G. Robles, L. A. Reina, J. M. González-Barahona, and S. D. Domínguez, "Women in free/libre/open source software: The situation in the 2010s," in *IFIP International Conference on Open Source Systems*. Springer, 2016, pp. 163–173.

[165] J. He, B. S. Butler, and W. R. King, "Team cognition: Development and evolution in software project teams," *Journal of Management Information Systems*, vol. 24, no. 2, pp. 261–292, 2007.

[166] D. Nafus, "'patches don't have gender': What is not open in open source software," *New Media & Society*, vol. 14, no. 4, pp. 669–683, 2012.

[167] M. E. Heilman, "Gender stereotypes and workplace bias," *Research in organizational Behavior*, vol. 32, pp. 113–135, 2012.

[168] E. H. Gorman and J. A. Kmec, "We (have to) try harder: Gender and required work effort in britain and the united states," *Gender & Society*, vol. 21, no. 6, pp. 828–856, 2007.

[169] M. E. Heilman, A. S. Wallen, D. Fuchs, and M. M. Tamkins, "Penalties for success: reactions to women who succeed at male gender-typed tasks." *Journal of applied psychology*, vol. 89, no. 3, p. 416, 2004.

[170] P. L. Roth, K. L. Purvis, and P. Bobko, "A meta-analysis of gender group differences for measures of job performance in field studies," *Journal of Management*, vol. 38, no. 2, pp. 719–739, 2012.

[171] H. W. Marsh, L. Bornmann, R. Mutz, H.-D. Daniel, and A. O'Mara, "Gender effects in the peer reviews of grant proposals: A comprehensive meta-analysis comparing traditional and multilevel approaches," *Review of Educational Research*, vol. 79, no. 3, pp. 1290–1326, 2009.

[172] S. Knobloch-Westerwick, C. J. Glynn, and M. Huge, "The matilda effect in science communication: an experiment on gender bias in publication quality perceptions and collaboration interest," *Science Communication*, vol. 35, no. 5, pp. 603–625, 2013.

[173] T. Nakagawa, Y. Kamei, H. Uwano, A. Monden, K. Matsumoto, and D. M. German, "Quantifying programmers' mental workload during program comprehension based on cerebral blood flow measurement: A controlled experiment," in *Companion Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 448–451.

[174] Y. Ikutani and H. Uwano, "Brain activity measurement during program comprehension with NIRS," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. IEEE, 2014, pp. 1–6.

[175] J. Siegmund, N. Peitek, C. Parnin, S. Apel, J. Hofmeister, C. Kästner, A. Begel, A. Bethmann, and A. Brechmann, "Measuring Neural Efficiency of Program Comprehension," in *Foundations of Software Engineering*, 2017, pp. 140–150. [Online]. Available: http://doi.acm.org/10.1145/3106237.3106268

[176] N. Peitek, J. Siegmund, C. Parnin, S. Apel, J. Hofmeister, and A. Brechmann, "Simultaneous Measurement of Program Comprehension with fMRI and Eye Tracking: A Case Study," in *Symposium on Empirical Software Engineering and Measurement*, 2018.

[177] J. Siegmund, C. Kästner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann, "Understanding understanding source code with functional magnetic resonance imaging," in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 378–389. [Online]. Available: https://doi-org.proxy.lib.umich.edu/10.1145/2568225.2568252

[178] I. Crk, T. Kluthe, and A. Stefik, "Understanding programming expertise: an empirical study of phasic brain wave changes," *Transactions on Computer-Human Interaction*, vol. 23, no. 1, p. 2, 2016.

[179] S. Lee, A. Matteson, D. Hooshyar, S. Kim, J. Jung, G. Nam, and H. Lim, "Comparing programming language comprehension between novice and expert programmers using EEG analysis," in *International Conference on Bioinformatics and Bioengineering*, Oct 2016, pp. 350–355.

[180] C. Parnin, "Subvocalization-toward hearing the inner thoughts of developers," in *International Conference on Program Comprehension*. IEEE, 2011, pp. 197–200.

[181] T. Fritz, A. Begel, S. C. Müller, S. Yigit-Elliott, and M. Züger, "Using psycho-physiological measures to assess task difficulty in software development," in *International Conference on Software Engineering*, 2014, pp. 402–413. [Online]. Available: http://doi.acm.org/10.1145/2568225.2568266

[182] S. Lee, D. Hooshyar, H. Ji, K. Nam, and H. Lim, "Mining biometric data to predict programmer expertise and task difficulty," *Cluster Computing*, pp. 1–11, 2017.

[183] L. Kaufmann, S. E. Vogel, G. Wood, C. Kremser, M. Schocke, L.-B. Zimmerhackl, and J. W. Koten, "A developmental fMRI study of nonsymbolic numerical and spatial processing," *Cortex*, vol. 44, no. 4, pp. 376–385, 2008.

[184] K. Landerl and C. Kölle, "Typical and atypical development of basic numerical skills in elementary school," *Journal of experimental child psychology*, vol. 103, no. 4, pp. 546–565, 2009.

[185] J. L. Booth and R. S. Siegler, "Numerical magnitude representations influence arithmetic learning," *Child development*, vol. 79, no. 4, pp. 1016–1031, 2008.

[186] J. Halberda, M. M. Mazzocco, and L. Feigenson, "Individual differences in non-verbal number acuity correlate with maths achievement," *Nature*, vol. 455, no. 7213, p. 665, 2008.

[187] J. A. Bugos, W. M. Perlstein, C. S. McCrae, T. S. Brophy, and P. H. Bedenbaugh, "Individualized piano instruction enhances executive functioning and working memory in older adults," *Aging and mental health*, vol. 11, no. 4, pp. 464–471, 2007.

[188] P. G. Simos, J. M. Fletcher, E. Bergman, J. Breier, B. Foorman, E. Castillo, R. Davis, M. Fitzgerald, and A. Papanicolaou, "Dyslexia-specific brain activation profile becomes normal following successful remedial training," *Neurology*, vol. 58, no. 8, pp. 1203–1213, 2002.

[189] B. McCandliss, I. L. Beck, R. Sandak, and C. Perfetti, "Focusing attention on decoding for children with poor reading skills: Design and preliminary tests of the word building intervention," *Scientific studies of reading*, vol. 7, no. 1, pp. 75–104, 2003.

[190] E. Dahlin, A. S. Neely, A. Larsson, L. Bäckman, and L. Nyberg, "Transfer of learning after updating training mediated by the striatum," *Science*, vol. 320, no. 5882, pp. 1510–1512, 2008.

[191] V. W. Berninger, T. L. Richards, P. S. Stock, R. D. Abbott, P. A. Trivedi, L. E. Altemeier, and J. R. Hayes, "fMRI activation related to nature of ideas generated and differences between good and poor writers during idea generation," in *BJEP Monograph Series II, Number 6-Teaching and Learning Writing*. British Psychological Society, 2009, vol. 77, no. 93, pp. 77–93.

[192] T. L. Richards, V. W. Berninger, and M. Fayol, "fMRI activation differences between 11-year-old good and poor spellers' access in working memory to temporary and long-term orthographic representations," *Journal of Neurolinguistics*, vol. 22, no. 4, pp. 327–353, 2009.

[193] C. Parnin, "A cognitive neuroscience perspective on memory for programming tasks," *Programming Interest Group*, p. 27, 2010.

[194] N. Faria, R. Silva, and J. L. Sobral, "Impact of data structure layout on performance," in *Parallel, Distributed and Network-Based Processing*, 2013, pp. 116–120.

[195] R. Williams, "A survey of data structures for computer graphics systems," in *Data Structures, Computer Graphics, and Pattern Recognition*. Elsevier, 1977, pp. 105–152.

[196] M. A.-M. Al-Shandawely, *Impacts of Data Structures and Algorithms on Multi-core Efficiency*. Skolan för datavetenskap och kommunikation, Kungliga Tekniska högskolan, 2010.

[197] V. Tkachenko, "How three fundamental data structures impact storage and retrieval," *https://dzone.com/articles/how-three-fundamental-data-structures-impact-stora*, Mar 2016.

[198] P. Oman and J. Hagemeister, "Metrics for assessing a software system's maintainability," in *Software Maintenance, 1992. Proceerdings., Conference on*. IEEE, 1992, pp. 337–344.

[199] Y. Aumann and M. A. Bender, "Fault tolerant data structures," in *Foundations of Computer Science*. IEEE, 1996, pp. 580–589.

[200] R. E. Strom and S. Yemini, "Typestate: A programming language concept for enhancing software reliability," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 1, pp. 157–171, 1986.

[201] H. Samet, *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1990.

[202] M. C. Linn and A. C. Petersen, "Emergence and characterization of sex differences in spatial ability: A meta-analysis," *Child development*, pp. 1479–1498, 1985.

[203] G. H. Glover, "Overview of functional magnetic resonance imaging," *Neurosurgery Clinics*, vol. 22, no. 2, pp. 133–139, 2011.

[204] C. Triantafyllou, R. Hoge, G. Krueger, C. Wiggins, A. Potthast, G. Wiggins, and L. Wald, "Comparison of physiological noise at 1.5 t, 3 t and 7 t and optimization of fMRI acquisition parameters," *Neuroimage*, vol. 26, no. 1, pp. 243–250, 2005.

[205] X. Xiao, H. Zhu, W.-J. Liu, X.-T. Yu, L. Duan, Z. Li, and C.-Z. Zhu, "Semi-automatic 10/20 identification method for MRI-free probe placement in transcranial brain mapping techniques," *Frontiers in Neuroscience*, vol. 11, no. 4, 2017.

[206] G. H. Klem, H. O. Lüders, H. Jasper, C. Elger *et al.*, "The ten-twenty electrode system of the international federation," *Electroencephalogr Clin Neurophysiol*, vol. 52, no. 3, pp. 3–6, 1999.

[207] T. C. Technologies, "10/20 system positioning," *https://www.trans-cranial.com/local/manuals/10_20_pos_man_v1_0_pdf.pdf*, 2012.

[208] M. Peters and C. Battista, "Applications of mental rotation figures of the shepard and metzler type and description of a mental rotation stimulus library," *Brain and cognition*, vol. 66, no. 3, pp. 260–264, 2008.

[209] Center for Diagnostic Imaging, "I'm getting an MRI, so what's a coil?" *https://www.mycdi.com/viewpoints/im_getting_an_mri_so_whats_a_coil_103*, Jan 2016.

[210] J. B. A. Maintz and M. A. Viergever, "A survey of medical image registration," *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998.

[211] J. Diedrichsen and R. Shadmehr, "Detecting and adjusting for artifacts in fMRI time series data," *NeuroImage*, vol. 27, no. 3, pp. 624 – 634, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811905003095

[212] J. W. Barker, A. Aarabi, and T. J. Huppert, "Autoregressive model based algorithm for correcting motion and serially correlated errors in fNIRS," *Biomedical optics express*, vol. 4, no. 8, pp. 1366–1379, 2013.

[213] R. L. Buckner, J. R. Andrews-Hanna, and D. L. Schacter, "The brain's default network," *Annals of the New York Academy of Sciences*, vol. 1124, no. 1, pp. 1–38, 2008. [Online]. Available: https://nyaspubs.onlinelibrary.wiley.com/doi/abs/10.1196/annals.1440.011

[214] S. Tak and J. C. Ye, "Statistical analysis of fNIRS data: A comprehensive review," *NeuroImage*, vol. 85, pp. 72–91, 2014, celebrating 20 Years of Functional Near Infrared Spectroscopy (fNIRS). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811913006538

[215] T. J. Huppert, "Commentary on the statistical properties of noise and its implication on general linear models in functional near-infrared spectroscopy," *Neurophotonics*, vol. 3, p. 010401, 03 2016.

[216] D. T. D, L. Chang, E. Caparelli, and T. Ernst, "Different activation patterns for working memory load and visual attention load," *Brain research*, vol. 1132, no. 1, pp. 158–165, 2007.

[217] J. Scholz, M. Klein, T. Behrens, and H. Johansen-Berg, "Training induces changes in white matter architecture," *Nature neuroscience*, vol. 12, no. 11, pp. 1370–1371, 2009.

[218] E. Maguire, D. Gadian, I. Johnsrude, C. Good, J. Ashburner, R. Frackowiak, and C. Frith, "Navigation-related structural change in the hippocampi of taxi drivers," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 8, pp. 4398–4403, 2000.

[219] P. A. Mabe and S. G. West, "Validity of self-evaluation of ability: A review and meta-analysis." *Journal of applied Psychology*, vol. 67, no. 3, p. 280, 1982.

[220] P. M. Podsakoff and D. W. Organ, "Self-reports in organizational research: Problems and prospects," *Journal of management*, vol. 12, no. 4, pp. 531–544, 1986.

[221] K. Atit, T. F. Shipley, and B. Tikoff, "Twisting space: are rigid and non-rigid mental transformations separate spatial skills?" *Cognitive processing*, vol. 14, no. 2, pp. 163–173, 2013.

[222] Q. Burke and Y. B. Kafai, "Programming & storytelling: opportunities for learning about coding & composition," in *Proceedings of the 9th international conference on interaction design and children*. ACM, 2010, pp. 348–351.

[223] C. Kelleher and R. Pausch, "Using storytelling to motivate programming," *Communications of the ACM*, vol. 50, no. 7, pp. 58–64, 2007.

[224] Q. Burke, "The markings of a new pencil: Introducing programming-as-writing in the middle school classroom." *Journal of Media Literacy Education*, vol. 4, no. 2, pp. 121–135, 2012.

[225] R. English and G. Edwards, "Programming as a writing activity," *Computing Teacher*, vol. 11, no. 6, pp. 46–47, 1984.

[226] Y. I. Sheline, D. M. Barch, J. M. Donnelly, J. M. Ollinger, A. Z. Snyder, and M. A. Mintun, "Increased amygdala response to masked emotional faces in depressed subjects resolves with antidepressant treatment: an fmri study," *Biological psychiatry*, vol. 50, no. 9, pp. 651–658, 2001.

[227] T. Nakao, A. Nakagawa, T. Yoshiura, E. Nakatani, M. Nabeyama, C. Yoshizato, A. Kudoh, K. Tada, K. Yoshioka, M. Kawamoto, O. Togao, and S. Kanba, "Brain activation of patients with obsessive-compulsive disorder during neuropsychological and symptom provocation tasks before and after symptom improvement: a functional magnetic resonance imaging study," *Biological psychiatry*, vol. 57, no. 8, pp. 901–910, 2005.

[228] S. J. Van Rooij, E. Geuze, M. Kennis, A. R. Rademaker, and M. Vink, "Neural correlates of inhibition and contextual cue processing related to treatment response in ptsd," *Neuropsychopharmacology*, vol. 40, no. 3, p. 667, 2015.

[229] M. Behroozi, A. Lui, I. Moore, D. Ford, and C. Parnin, "Dazed: measuring the cognitive load of solving technical interview problems at the whiteboard," in *International Conference on Software Engineering: New Ideas and Emerging Results (ICSE NIER)*, 2018, pp. 93–96. [Online]. Available: https://doi.org/10.1145/3183399.3183415

[230] R. T. Kellogg, "Training writing skills: A cognitive developmental perspective," *Journal of Writing Research*, vol. 1, no. 1, pp. 1–26, 2008.

[231] L. W. Gregg and E. R. Steinberg, *Cognitive processes in writing*. Routledge, 2016.

[232] T. L. Richards, V. W. Berninger, P. Stock, L. Altemeier, P. Trivedi, and K. R. Maravilla, "Differences between good and poor child writers on fMRI contrasts for writing newly taught and highly practiced letter forms," *Reading and Writing*, vol. 24, no. 5, pp. 493–516, 2011.

[233] D. Arnow and G. Weiss, "Turing's craft," in *https://www.turingscraft.com/*, 2019.

[234] D. Arnow and O. Barshay, "On-line programming examinations using web to teach," in *Conference on Innovation and Technology in Computer Science Education*, 1999, pp. 21–24. [Online]. Available: http://doi.acm.org/10.1145/305786.305835

[235] V. Sarina and I. K. Namukasa, "Nonmath analogies in teaching mathematics," *Procedia-Social and Behavioral Sciences*, vol. 2, no. 2, pp. 5738–5743, 2010.

[236] R. G. Cuya, G. Nivera, and E. C. Fortes, "The use of non-math analogies in teaching mathematics," *The Normal Lights*, vol. 11, no. 1, 2017.

[237] Psychology Software Tools, Inc., "E-Prime," https://pstnet.com/products/e-prime/.

[238] M. Snejbjerg Jensen, O. Heggli, P. Mota, and P. Vuust, "A low-cost MRI compatible keyboard," in *International Conference on New Interfaces for Musical Expression*, 2017.

[239] Y. Higashiyama, K. Takeda, Y. Someya, Y. Kuroiwa, and F. Tanaka, "The neural basis of typewriting: A functional MRI study," *PLOS ONE*, vol. 10, no. 7, pp. 1–20, 07 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0134131

[240] G. A. James, G. He, and Y. Liu, "A full-size MRI-compatible keyboard response system," *Neuroimage*, vol. 25, no. 1, pp. 328–31, Mar. 2005.

[241] C. M. Bennett, M. Miller, and G. L. Wolford, "Neural correlates of interspecies perspective taking in the post-mortem atlantic salmon: An argument for proper multiple comparisons correction," *NeuroImage*, vol. 47, Jul. 2009.

[242] Wellcome Trust Centre for Neuroimaging, "Statistical parametric mapping," in *http://www.fil.ion.ucl.ac.uk/spm/*, Aug. 2016.

[243] S. Ogawa, T. M. Lee, A. R. Kay, and D. W. Tank, "Brain magnetic resonance imaging with contrast dependent on blood oxygenation," *Proceedings of the National Academy of Sciences*, vol. 87, no. 24, pp. 9868–9872, 1990. [Online]. Available: http://www.pnas.org/content/87/24/9868.abstract

[244] J. Diedrichsen and R. Shadmehr, "Detecting and adjusting for artifacts in fMRI time series data," *NeuroImage*, vol. 27, no. 3, pp. 624–634, 2005.

[245] H. R. Heekeren, S. Marrett, P. A. Bandettini, and L. G. Ungerleider, "A general mechanism for perceptual decision-making in the human brain," *Nature*, vol. 431, no. 7010, p. 859, 2004.

[246] S.-n. Yang, H. Hwang, J. Ford, and S. Heinen, "Supplementary eye field activity reflects a decision rule governing smooth pursuit but not the decision," *Journal of neurophysiology*, vol. 103, no. 5, pp. 2458–2469, 2010.

[247] M. Koenigs, A. K. Barbey, B. R. Postle, and J. Grafman, "Superior parietal cortex is critical for the manipulation of information in working memory," *Journal of Neuroscience*, vol. 29, no. 47, pp. 14 980–14 986, 2009.

[248] M. Corbetta, G. L. Shulman, F. M. Miezin, and S. E. Petersen, "Superior parietal cortex activation during spatial attention shifts and visual feature conjunction," *Science*, vol. 270, no. 5237, pp. 802–805, 1995.

[249] D. Kemmerer, "Word classes in the brain: Implications of linguistic typology for cognitive neuroscience," *Cortex*, vol. 58, pp. 27–51, 2014.

[250] M. V. Peelen, D. Romagno, and A. Caramazza, "Independent representations of verbs and actions in left lateral temporal cortex," *Journal of cognitive neuroscience*, vol. 24, no. 10, pp. 2096–2107, 2012.

[251] R. Campbell, "The processing of audio-visual speech: empirical and neural bases," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 363, no. 1493, pp. 1001–1010, 2007.

[252] A. Mechelli, G. W. Humphreys, K. Mayall, A. Olson, and C. J. Price, "Differential effects of word length and visual contrast in the fusiform and lingual gyri during," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 267, no. 1455, pp. 1909–1913, 2000.

[253] A. Flinker, A. Korzeniewska, A. Y. Shestyuk, P. J. Franaszczuk, N. F. Dronkers, R. T. Knight, and N. E. Crone, "Redefining the role of broca's area in speech," *Proceedings of the National Academy of Sciences*, vol. 112, no. 9, pp. 2871–2875, 2015.

[254] S. Zhang and R. L. Chiang-shan, "Functional connectivity mapping of the human precuneus by resting state fmri," *Neuroimage*, vol. 59, no. 4, pp. 3548–3562, 2012.

[255] R. H. Grabner, A. Ischebeck, G. Reishofer, K. Koschutnig, M. Delazer, F. Ebner, and C. Neuper, "Fact learning in complex arithmetic and figural-spatial tasks: The role of the angular gyrus and its relation to mathematical competence," *Human brain mapping*, vol. 30, no. 9, pp. 2936–2952, 2009.

[256] M. L. Seghier, "The angular gyrus: multiple functions and multiple subdivisions," *The Neuroscientist*, vol. 19, no. 1, pp. 43–61, 2013.

[257] E. Dijkstra, "How do we tell truths that might hurt?" in *Selected Writings on Computing: A Personal Perspective*. Springer, 1982, pp. 129–131.

[258] T. Busjahn, R. Bednarik, A. Begel, M. Crosby, J. H. Paterson, C. Schulte, B. Sharif, and S. Tamm, "Eye movements in code reading: Relaxing the linear order," in *International Conference on Program Comprehension*, 2015, pp. 255–265.

[259] A. Bosu, M. Greiler, and C. Bird, "Characteristics of useful code reviews: An empirical study at microsoft," in *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. IEEE, 2015, pp. 146–156.

[260] N. Kennedy, "How google does web-based code reviews with mondrian," 2006.

[261] M. Welsh, "My love affair with code reviews," http://matt-welsh.blogspot.com/2012/02/my-love-affair-with-code-reviews.html, 2012, [Online; accessed 4-September-2019].

[262] A. Tsotsis, "Meet phabricator, the witty code review tool built inside facebook," *City*, 2006.

[263] yeeguy, "How Facebook Ships Code," https://framethink.wordpress.com/2011/01/17/how-facebook-ships-code/, 2011, [Online; accessed 4-September-2019].

[264] A. Bosu and J. C. Carver, "Impact of peer code review on peer impression formation: A survey," in *Empirical Software Engineering and Measurement*, 2013.

[265] P. C. Rigby, D. M. German, and M.-A. Storey, "Open source software peer review practices: a case study of the apache server," in *Proceedings of the 30th international conference on Software engineering*. ACM, 2008, pp. 541–550.

[266] J. Marlow, L. Dabbish, and J. Herbsleb, "Impression formation in online peer production: activity traces and personal profiles in github," in *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 2013, pp. 117–128.

[267] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, and V. Filkov, "Gender and tenure diversity in github teams," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM, 2015, pp. 3789–3798.

[268] S. Sarkar and C. Parnin, "Characterizing and predicting mental fatigue during programming tasks," in *Emotion Awareness in Software Engineering*, 2017.

[269] T. Camp, W. DuBow, D. Levitt, L. J. Sax, V. Taylor, and C. Lewis, "The new NSF requirement for broadening participation in computing (BPC) plans: Community advice and resources," in *Computer Science Education*, 2019, pp. 332–333.

[270] J. Cohen, "11 proven practices for more effective, efficient peer code review," https://www.ibm.com/developerworks/rational/library/11-proven-practices-for-peer-review/index.html, January 2011.

[271] David Meyer, "Amazon Reportedly Killed an AI Recruitment System Because It Couldn't Stop the Tool from Discriminating Against Women," https://https://fortune.com/2018/10/10/amazon-ai-recruitment-bias-women-sexist/.

[272] D. Wakabayashi, "Google finds it's underpaying many men as it addresses wage equity," https://www.nytimes.com/2019/03/04/technology/google-gender-pay-gap.html, March 2019.

[273] G. Robles, L. Arjona Reina, A. Serebrenik, B. Vasilescu, and J. M. González-Barahona, "Floss 2013: A survey dataset about free software contributors: challenges for curating, sharing, and combining," in *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM, 2014, pp. 396–399.

[274] S. Hoogendoorn, H. Oosterbeek, and M. Van Praag, "The impact of gender diversity on the performance of business teams: Evidence from a field experiment," *Management Science*, vol. 59, no. 7, pp. 1514–1528, 2013.

[275] S. Zweben and B. Bizot, "2017 CRA Taulbee Survey," *Computing Research News*, vol. 30, no. 5, pp. 1–47, 2018.

[276] M. Monperrus, "Automatic software repair: A bibliography," *ACM Comput. Surv.*, vol. 51, no. 1, Jan. 2018. [Online]. Available: https://doi-org.proxy.lib.umich.edu/10.1145/3105906

[277] C. Goues, S. Forrest, and W. Weimer, "Current challenges in automatic software repair," *Software Quality Journal*, vol. 21, no. 3, p. 421–443, Sep. 2013. [Online]. Available: https://doi-org.proxy.lib.umich.edu/10.1007/s11219-013-9208-0

[278] A. Marginean, J. Bader, S. Chandra, M. Harman, Y. Jia, K. Mao, A. Mols, and A. Scott, "SapFix: Automated end-to-end repair at scale," in *International Conference on Software Engineering: Software Engineering in Practice*, 2019.

[279] S. O. Haraldsson, J. R. Woodward, A. E. I. Brownlee, and K. Siggeirsdottir, *Fixing Bugs in Your Sleep: How Genetic Improvement Became an Overnight Success*, 2017. [Online]. Available: https://doi-org.proxy.lib.umich.edu/10.1145/3067695.3082517

[280] C. Le Goues, M. Dewey-Vogt, S. Forrest, and W. Weimer, "A systematic study of automated program repair: Fixing 55 out of 105 bugs for $8 each," in *International Conference on Software Engineering*, 2012.

[281] T. J. Ryan, G. M. Alarcon, C. Walter, R. Gamble, S. A. Jessup, A. Capiola, and M. D. Pfahler, "Trust in automated software repair," in *International Conference on Human-Computer Interaction*. Springer, 2019, pp. 452–470.

[282] S. Urli, Z. Yu, L. Seinturier, and M. Monperrus, "How to design a program repair bot? insights from the Repairnator project," in *International Conference on Software Engineering: Software Engineering in Practice*, 2018. [Online]. Available: https://doi-org.proxy.lib.umich.edu/10.1145/3183519.3183540

[283] J. G. Altonji and R. M. Blank, "Race and gender in the labor market," *Handbook of labor economics*, vol. 3, pp. 3143–3259, 1999.

[284] S. Beyer, "Gender differences in the accuracy of self-evaluations of performance." *Journal of personality and social psychology*, vol. 59, no. 5, p. 960, 1990.

[285] J. D. Ivory, "Still a man's game: Gender representation in online reviews of video games," *Mass Communication & Society*, vol. 9, no. 1, pp. 103–114, 2006.

[286] D. M. Johnson and D. H. Roen, "Complimenting and involvement in peer reviews: Gender variation," *Language in society*, vol. 21, no. 1, pp. 27–57, 1992.

[287] P. Tse and K. Hyland, "'robot kung fu': Gender and professional identity in biology and philosophy reviews," *Journal of Pragmatics*, vol. 40, no. 7, pp. 1232–1248, 2008.

[288] Z. Cattaneo, G. Mattavelli, E. Platania, and C. Papagno, "The role of the prefrontal cortex in controlling gender-stereotypical associations: a tms investigation," *NeuroImage*, vol. 56, no. 3, pp. 1839–1846, 2011.

[289] J. S. Beer, M. Stallen, M. V. Lombardo, K. Gonsalkorale, W. A. Cunningham, and J. W. Sherman, "The quadruple process model approach to examining the neural underpinnings of prejudice," *Neuroimage*, vol. 43, no. 4, pp. 775–783, 2008.

[290] S. Quadflieg, D. J. Turk, G. D. Waiter, J. P. Mitchell, A. C. Jenkins, and C. N. Macrae, "Exploring the neural correlates of social stereotyping," *Journal of Cognitive Neuroscience*, vol. 21, no. 8, pp. 1560–1570, 2009.

[291] M. Gozzi, V. Raymont, J. Solomon, M. Koenigs, and J. Grafman, "Dissociable effects of prefrontal and anterior temporal cortical lesions on stereotypical gender attitudes," *Neuropsychologia*, vol. 47, no. 10, pp. 2125–2132, 2009.

[292] Q. Luo, M. Nakic, T. Wheatley, R. Richell, A. Martin, and R. J. R. Blair, "The neural basis of implicit moral attitude—an iat study using event-related fmri," *Neuroimage*, vol. 30, no. 4, pp. 1449–1457, 2006.

[293] X. Jiang, E. Rosen, T. Zeffiro, J. VanMeter, V. Blanz, and M. Riesenhuber, "Evaluation of a shape-based model of human face discrimination using fmri and behavioral techniques," *Neuron*, vol. 50, no. 1, pp. 159–172, 2006.

[294] W. A. Cunningham, J. J. Van Bavel, and I. R. Johnsen, "Affective flexibility: evaluative processing goals shape amygdala activity," *Psychological Science*, vol. 19, no. 2, pp. 152–160, 2008.

[295] A. M. Chekroud, J. A. Everett, H. Bridge, and M. Hewstone, "A review of neuroimaging studies of race-related prejudice: does amygdala response reflect threat?" *Frontiers in Human Neuroscience*, vol. 8, p. 179, 2014.

[296] H. Uwano, M. Nakamura, A. Monden, and K.-i. Matsumoto, "Analyzing individual performance of source code review using reviewers' eye movement," in *Eye Tracking Research Applications*, 2006. [Online]. Available: https://doi-org.proxy.lib.umich.edu/10.1145/1117309.1117357

[297] B. Sharif, M. Falcone, and J. I. Maletic, "An eye-tracking study on the role of scan time in finding source code defects," in *Symposium on Eye Tracking Research and Applications*, 2012. [Online]. Available: https://doi-org.proxy.lib.umich.edu/10.1145/2168556.2168642

[298] A. Begel and H. Vrzakova, "Eye movements in code review," in *Proceedings of the Workshop on Eye Movements in Programming*, 2018. [Online]. Available: https://doi-org.proxy.lib.umich.edu/10.1145/3216723.3216727

[299] D. S. Ma, J. Correll, and B. Wittenbrink, "The chicago face database: A free stimulus set of faces and norming data," *Behavior research methods*, vol. 47, no. 4, pp. 1122–1135, 2015.

[300] J. J. Dolado, M. C. Otero, and M. Harman, "Equivalence hypothesis testing in experimental software engineering," *Software Quality Journal*, vol. 22, no. 2, pp. 215–238, 2014.

[301] P. Grimm, "Social desirability bias," *Wiley international encyclopedia of marketing*, 2010.

[302] A. G. Greenwald, D. E. McGhee, and J. L. Schwartz, "Measuring individual differences in implicit cognition: the implicit association test." *Journal of personality and social psychology*, vol. 74, no. 6, p. 1464, 1998.

[303] B. A. Nosek, A. G. Greenwald, and M. R. Banaji, "Understanding and using the implicit association test: Ii. method variables and construct validity," *Personality and Social Psychology Bulletin*, vol. 31, no. 2, pp. 166–180, 2005.

[304] D. A. Magezi, "Linear mixed-effects models for within-participant psychology experiments: an introductory tutorial and free, graphical user interface (lmmgui)," *Frontiers in psychology*, vol. 6, p. 2, 2015.

[305] J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins, "The aligned rank transform for nonparametric factorial analyses using only anova procedures," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2011, pp. 143–146.

[306] L. Beckwith, D. Inman, K. Rector, and M. Burnett, "On to the real world: Gender and self-efficacy in excel," in *Proceeding of the 2007 Symposium on Visual Languages and Human-Centric Computing*. IEEE, 2007, pp. 119–126.

[307] N. Subrahmaniyan, L. Beckwith, V. Grigoreanu, M. Burnett, S. Wiedenbeck, V. Narayanan, K. Bucht, R. Drummond, and X. Fern, "Testing vs. code inspection vs. what else?: Male and female end users' debugging strategies," in *Proceedings of the 2008 SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 617–626. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357153

[308] M. Vorvoreanu, L. Zhang, Y.-H. Huang, C. Hilderbrand, Z. Steine-Hanson, and M. Burnett, "From gender biases to gender-inclusive design: An empirical investigation," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019, p. 53.

[309] Z. Sharafi, Z. Soh, and Y.-G. Guéhéneuc.

[310] R. Krueger, Y. Huang, X. Liu, T. Santander, W. Weimer, and K. Leach, "Neurological divide: An fmri study of prose and code writing," in *International Conference on Software Engineering*, 2020.

[311] F. Long and M. Rinard, "Automatic patch generation by learning correct code," in *Principles of Programming Languages*, 2016. [Online]. Available: https://doi.org/10.1145/2837614.2837617

[312] R. van Tonder and C. Le Goues, "Towards s/engineer/bot: Principles for program repair bots," in *2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE)*, May 2019, pp. 43–47.

[313] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. E. Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," *IEEE Transactions on Software Engineering*, vol. 28, no. 8, pp. 721–734, Aug. 2002. [Online]. Available: https://doi.org/10.1109/TSE.2002.1027796

[314] J. R. Shapiro and S. L. Neuberg, "From stereotype threat to stereotype threats: Implications of a multi-threat framework for causes, moderators, mediators, consequences, and interventions," *Personality and Social Psychology Review*, vol. 11, no. 2, pp. 107–130, 2007.

[315] R. Krueger, Y. Huang, X. Liu, T. Santander, Westley, and K. Leach, "Neurological divide: An fMRI study of prose and code writing," ser. ICSE '20. Seoul, South Korea: IEEE Press, 2020.

[316] Y. Huang, K. Leach, **Z. Sharafi**, N. McKay, T. Santander, and W. Weimer, "Investigating gender bias and differences in code review: Using medical imaging and eye-tracking,"

in *International Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM/SIGSOFT, 2020.

[317] Z. Sharafi, Y. Huang, K. Leach, and W. Weimer, "Toward an objective measure of developers' cognitive activities," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 30, no. 3, pp. 1–40, 2021.